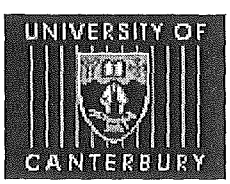
Implementation of a Direct Torque Controller using a Three-Level Inverter for an Electric Vehicle Application

A Thesis submitted in partial fulfilment of the requirements for the Degree of Master of Engineering (Electrical and Electronic)

Jonathan Charles Trounce, BE (Hons)

November 2001



University of Canterbury
Christchurch, New Zealand

Abstract

The University of Canterbury has been involved in the research and development of electric vehicles since 1974. These electric vehicles use induction motors that are designed with a low inductance such that they can run at high speed and therefore have a high power to weight ratio. The inverter switching can cause high current ripple due to the low inductance of these motors. A three-level inverter has been previously developed for electric vehicle applications that produces an improved voltage waveform and therefore has reduced current distortion.

This thesis describes the development of an induction motor torque controller for a three-level inverter in an electric vehicle application. Four different induction motor torque control schemes are considered. These are field oriented control, direct torque control (DTC), minimal torque ripple DTC, and DTC using space vector modulation. All four of the control schemes are simulated using MATLAB and Simulink, and DTC using space vector modulation is chosen based on the simulation results. DTC using space vector modulation is shown to have the low steady-state torque ripple, flux ripple, and current distortion that is characteristic of space vector modulation and the fast transient performance that is characteristic of direct torque control. The implementation of DTC using space vector modulation that is used is described in detail.

DTC using space vector modulation is implemented on a custom-built embedded controller based on a TMS320VC33 DSP and a XC4020XLA FPGA. The implementation of the control algorithms and a number of supporting software components are also described.

The implemented torque controller is tested against a 15hp, 400Hz, 200V induction motor. Results are presented that show the performance of the torque controller and the three-level space vector modulator. The three-level space vector modulator is shown to produce less current distortion at low modulation indices. The torque produced by the controller during steady-state operation differs from the reference torque, but this error is relatively constant over the range of speeds that are tested. The estimated response to step-changes in the torque and flux references are shown to be almost instantaneous.

Acknowledgements

There have been a number of people that have provided assistance with this project. I would especially like to thank my supervisor, Dr Simon Round, for his support, inspiration, and always being there to offer advice when things weren't working like they should.

I would like to thank the various members of the Power Electronics Research Group, particularly Hamish Laird for his many ideas and words of wisdom on many aspects of this project. I've learnt a lot from the discussions that I've had with Hamish and greatly appreciate the time that he has given me.

Secondly, I'd like to thank the many technicians in the Electrical and Electronic Engineering Department that have provided assistance. These include the technicians of the Power Electronics and Electrical Machines labs, Ron Battersby and Ken Smart who helped with tools, equipment, and the experimental set-up. I'd also like to thank the Electrical Workshop technicians, especially Nick Smith who mounted the fine pitch 144pin and 160pin devices on the PCBs. His patience with the work-around for my mistake with the incorrect PCB footprint for the 160pin FPGA is certainly appreciated. And like also like to thank the Mechanical Workshop technician, Peter Lambert, for the construction of the test jig needed to mechanically couple the motors for testing.

Finally, I'd like to thank the various companies that have provided hardware for this project, especially PDL Electronics that provided the Microdrive Elite ME-22.5 motor drive that was used for testing. I'm also grateful for the sample component programmes of Maxim and Texas Instruments that provided some of the components used in the hardware.

Related Publications

Trounce, J.C., Round, S.D. and Duke, R.M. (2001), "Comparison by Simulation of Three-Level Induction Motor Torque Control Schemes for Electric Vehicle Applications", *Proc. of International Power Engineering Conference*, Singapore, May 2001, vol. 1, pp. 294-299

Trounce, J.C., Round, S.D. and Duke, R.M. (2001), "Evaluation of direct torque control using space vector modulation for electric vehicle applications", *Proc. of Australasian University Power Engineering Conference*, Perth, Australia, Sept. 2001, vol. 1, pp. 292-297

Table of Contents

Table of Figures

Table of Tables

1. Introduction

The University of Canterbury has been involved in the research and development of electric vehicles since the oil crisis of 1974 [Byers, 1983]. The main purpose was to provide a suitable test-bed to demonstrate drive systems, control systems, and technologies related to electric vehicles. The first vehicle, the 'Mark I', was completed in 1976 as an extension into research on induction motor control, which was rarely used in electric vehicles at that time. The 'Mark I' vehicle was driven by two 2.2kW induction motors, which were rewound to operate at 90V (at 50Hz) and at up 150Hz. The motors where powered from twenty 12V lead-acid batteries (240V total) via a SCR inverter. This first vehicle had a range of 40km on a single charge when driven at a constant 50km/hr [Harman, 1992].

A second electric vehicle, the 'Mark II', was developed over the period of 1976 to 1982 based on a 1962 Austin A40 Farina body and using the same electrical components as the 'Mark I'. This vehicle was able to achieve a range of 60km at a constant 65km/hr, due to its improved drag co-efficient, and had a top speed of 80km. A Bipolar Junction Transistor (BJT) inverter was developed in 1993 to replace the original SCR based inverter [Williams, 1993]. The new BJT inverter was originally driven by a modified GEC motor speed controller but was later replaced by a controller from a commercial PDL motor drive.

All of the controllers used in these two electric vehicles were based on the slip-demand principle of induction torque motor control. To make the electric vehicle drive like a standard vehicle with an internal combustion engine, torque control is used rather than speed control. If the vehicle is going too fast, the driver releases the accelerator slightly, less torque is produced, and the vehicle slows down. Varying the supply frequency provides rough control of the induction motor speed. The motor will run slightly slower than the supply frequency depending on the mechanical load. The difference between the supply frequency and the actual motor speed is called the slip speed. Varying the slip speed will control the torque produced by the induction motor.

There are a number of problems with the slip-demand principle of torque control. As the slip speed is increased, the torque produced will increase up to a certain limit, above which the torque will start to decrease. Since the torque-slip characteristic of the motor changes with operating conditions, a conservative limit is used to ensure that the slip speed for maximum torque is never exceeded. This results in an under-utilisation of the induction motor torque capability and the inverter current capability. Improved torque control performance can be obtained by controlling the magnetic fields within the induction motor. This is relatively difficult to achieve and involves dividing the supply current into torque and flux producing components and controlling each component separately.

In March 1999, the development of an improved inverter was completed [Li, 1999]. It was intended that two of these inverters would be constructed such that two separate torque controllers could control each motor independently. These new inverters were to replace the single BJT inverter in the 'Mark II' electric vehicle. This new inverter was based on Insulated Gate Bipolar Transistors (IGBTs) that can switch faster and don't require the high base drive currents of BJTs. An additional advantage of this new inverter was a more sophisticated design that enabled it to produce three-levels of phase voltage compared to the two-levels of a standard inverter. This new three-level IGBT inverter could produce a much better approximation to the desired sinusoidal output voltage.

The completed three-level IGBT inverter was only tested driving a motor under open-loop conditions as no suitable torque controller was available that could generate the more complex switching patterns that were required. This project started shortly after development of the inverter was completed. The aim is to develop a suitable controller that would allow the induction motor torque to be controlled using the three-level inverter. Shortly after this project started, work commenced on a separate project to build a third electric vehicle.

The development of the third electric vehicle, called the 'EV3', begun in December of 1999 with the purchase of a 1992 Toyota MR2. The EV3 is intended to have good acceleration and a top speed of over 120km/hr to make it drive like a sports car. The EV3 is to be driven by an AC42 induction motor from Solectria Corporation. This is a 130Hz, 4-pole motor that has a peak rating of 70kW, 140Nm, 250Arms (42Nm, 21kW continuous

rating) and will run at up to 10,000RPM. It will be powered from 26 lead-acid batteries (312V DC total) and driven by a custom-built two-level 250A rms IGBT inverter.

1.1 Project Outline

This project combines an investigation into two areas, torque control using a three-level inverter and torque control of high-speed induction motors such as the 10,000RPM motor obtained for the new EV3 electric vehicle.

An important consideration in the selection of the motor for the electric vehicle is its weight and efficiency. A heavy motor will increase the overall weight of the vehicle, resulting in lower acceleration and reduced overall performance. Induction motors are a good choice as they are relatively cheap, lightweight and robust. High-speed motors typically have a higher power to weight ratio. The 130Hz AC42 motor selected for the EV3 vehicle weighs 60kg. A 50Hz induction motor with the same continuous power rating would weigh around 200kg, which is over three times as much.

To keep the supply voltage at an acceptable level, high-speed induction motors such as the AC42 are designed with a low inductance. The induction motor acts as a low-pass filter when driven by an inverter. The sinusoidal-modulated square wave voltage output of the inverter is filtered by the motor inductance such that an approximately sinusoidal current is supplied. The lower the inductance of the motor, the less filtering it provides, and the higher the current ripple due to the switching. These high frequency current components cause a significant amount of eddy current and hysteresis losses and therefore reduce the overall efficiency of the system.

A second problem with a high-speed induction motor such as the AC42 is the high excitation frequency that is required. To run the AC42 motor at 10,000RPM requires an excitation frequency of around 340Hz. The closer the fundamental frequency gets to the switching frequency, the more distorted the sinusoidal voltage that is produced. To maintain rated flux in an induction motor, the amplitude to frequency ratio of the applied voltage is kept approximately constant as the motor speed varies. Since the electric vehicle has minimal gearing, the motor needs to be operated over a wide speed range. At

500RPM, the motor is running at 5% of its maximum speed and therefore the applied voltage amplitude needs to be about 5% of the rated voltage. It is difficult to accurately generate a voltage down to such low levels using a standard two-level inverter.

High-speed induction motors have a lower inductance that results in increased current ripple. A three-level inverter offers some advantages over conventional two-level inverters when driving these motors. It generates an improved voltage that will produce less current distortion and switching ripple and it can also generate low amplitude voltages more accurately.

This project aims to implement a suitable torque controller for a high-speed electric vehicle induction motor driven from a three-level inverter. A second-hand 15hp, 200V, 400Hz, 12,000RPM induction motor is used for initial testing instead of the new 70kW, 10,000RPM electric vehicle motor. This is driven using a previously constructed three-level IGBT inverter [Li, 1999].

1.2 Thesis Outline

Chapter 2 of this thesis explains the structure of the three-level inverter and describes its advantages over a standard two-level inverter. It also develops a voltage model of the three-level inverter and describes a mathematical model of the induction motor. These models are then used as the basis for a description of three different torque control schemes. These are field-oriented control, direct torque control (DTC), and minimal torque ripple DTC. A fourth torque control scheme, DTC using space vector modulation, is described in detail in chapter 3 as it is the scheme that is eventually chosen. Its operation is described and all the details required for a practical implementation are presented.

Pulse width modulation (PWM) is normally used to generate the inverter switching patterns in order to synthesise a desired output voltage. Space vector modulation is a form of PWM that produces a high quality voltage waveform and allows a high utilisation of the inverter by producing a larger maximum output voltage. An implementation of space vector modulation for a three-level inverter is described in chapter 4. Due to

practical limitations of the inverter, small delays need to be inserted into the switching pattern. These introduce an error into the inverter's output voltage that affects the torque controller. Chapter 4 describes how this error is calculated for a three-level inverter such that the torque controller can compensate for it.

The four torque control schemes considered by this thesis are simulated using MATLAB and Simulink. The results are presented in chapter 5, along with a comparison of the advantages and disadvantages of each of the four torque control schemes. Chapter 5 explains why DTC using space vector modulation is chosen.

Chapters 6 and 7 describe the implementation of various hardware and software components such that the chosen torque control scheme can be implemented and evaluated experimentally. A sophisticated embedded hardware platform is created that is suitable for use in an electric vehicle. This hardware is based on a digital signal processor (DSP) and a high-density re-programmable logic device called a Field Programmable Gate Array (FPGA). Chapter 7 describes the software for the DSP that implements the torque controller and also several other components of supporting software.

The results of testing the implemented torque controller with a 15hp, 200V, 400Hz, 12,000RPM induction motor are described in chapter 8. The performance of the three-level space vector modulator and the steady state and transient performance of the torque controller is presented. These results are compared with the simulation results presented in chapter 5. Finally, chapter 9 describes the future work required to improve the system and chapter 10 concludes this thesis.

2. Torque Control using a Three-Level Inverter

This chapter provides an introduction to three-level inverters and the various induction motor torque control schemes that are possible. Section 2.1 provides a detailed description of the three-level inverter that is used for this project and develops an inverter model. Section 2.2 presents a detailed mathematical model of the induction motor. This model, along with the inverter model, is used as a basis for the explanation of the various torque control schemes that are presented in Section 2.3. The torque control schemes presented in Section 2.3 serve as a comparison to the actual torque control scheme that is used in the project, which is described in detail in chapter 3.

2.1 Three-Level Inverters

A conventional three-phase, two-level, six-switch inverter is able to switch each phase between two voltages. These are the positive and negative rails of the DC bus. A three-level inverter is able to produce a phase voltage consisting of three voltage levels (positive, negative, and zero).

The following sections provide a comparison between two-level and three-level inverters and detailed information on the three-level inverter used for this project.

2.1.1 Comparison with Two-Level Inverters

By producing an output voltage having more levels, the inverter can better approximate the required sinusoidal output voltage. This is demonstrated by Figure 2.1 which shows a comparison of the phase-to-neutral and the phase-to-phase voltages produced by two-level and three-level inverters. A better approximation of a sinusoidal voltage can be achieved for the same device switching frequency, which results in reduced current harmonics in the load. For an induction motor these current harmonics cause increased copper losses, iron losses, torque pulsations, noise and mechanical stresses. A three-level inverter causes lower losses in the induction motor and therefore results in a more efficient system.

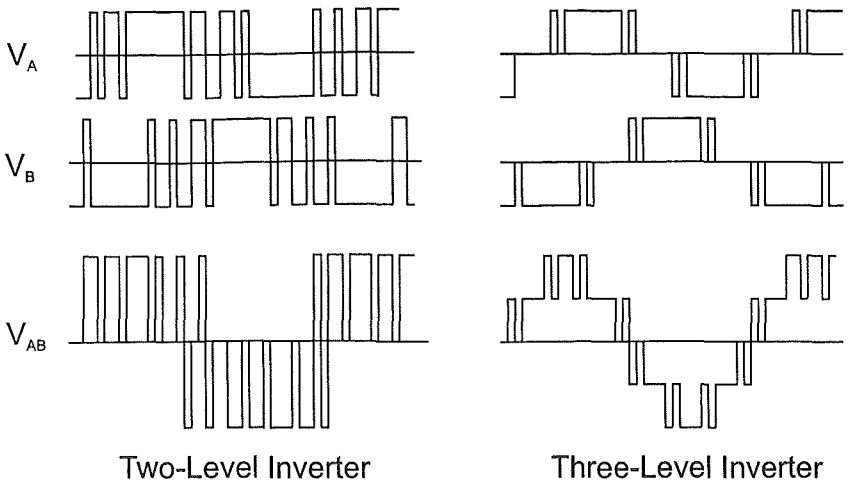


Figure 2.1: Comparison of output voltages for two-level and three-level inverters

Three-level inverters can also offer other advantages, such as an increased output voltage and reduced dv/dt switching stresses [Tolbert, Peng et al., 1998]. Since three-level inverters are designed to split the voltage across a number of switching devices with no voltage sharing problems, it is possible to synthesise higher output voltages using multiple devices of a lower voltage rating. The increased number of output voltage levels also reduces the voltage change (dv/dt) for each commutation. This results in less dv/dt stress on the switching devices and motor windings, and also results in reduced electro-magnetic interference (EMI).

The disadvantage of three-level inverters is their increased complexity and therefore cost. Compared to a two-level inverter, a three-level inverter requires at least twice as many switching devices, each with associated gate drive and protection circuits. Additional capacitors and/or diodes are also typically required. There is also the additional complexity in the generation of the switching patterns required to control the larger number of devices.

2.1.2 Inverter Structure

There are a number of techniques that can be used to obtain multiple levels of output voltage [Suh, Sinha et al., 1998]. These include various techniques to connect each phase to various tapings on the DC bus, multiple two-level inverters with interconnecting inductors, and inverters constructed using multiple cascaded H-bridges for each phase.

This project uses a three-level neutral point clamped (NPC) configuration first published in 1981 [Nabae, Takahashi et al., 1981]. A prototype neutral point clamped IGBT inverter with a 600V, 150A (peak) rating has been previously developed [Li, 1999]. A diagram of the NPC configuration is shown in Figure 2.2. Each phase consists of four IGBTs with associated freewheeling diodes and two additional clamping diodes.

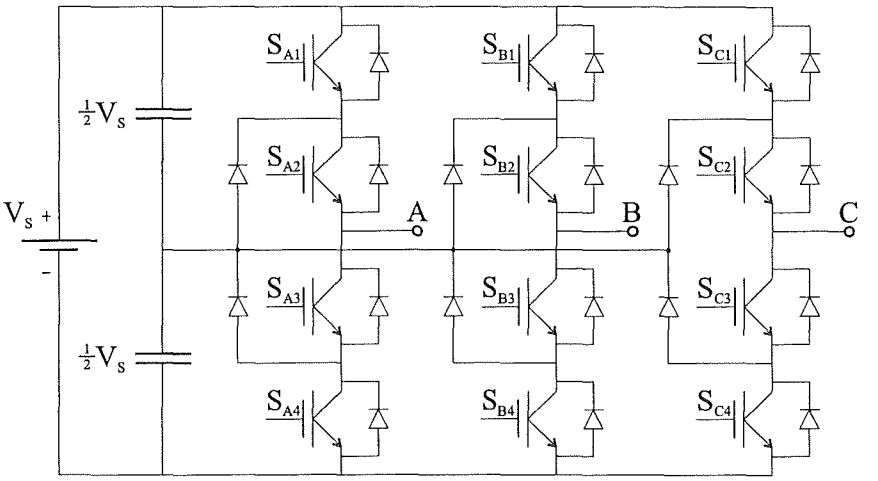


Figure 2.2: Three-Level Neutral Point Clamped Inverter

The IGBTs are switched in pairs consisting of (S_1, S_2), (S_2, S_3), and (S_3, S_4). When the upper pair (S_1, S_2) are turned on, the output is connected to the positive rail of the DC bus. When the lower pair (S_3, S_4) are turned on, the output is connected to the negative rail of the DC bus. When the two inner IGBTs (S_2, S_3) are turned on, the output is clamped to the bus neutral point through one of the inner IGBTs and its associated clamping diode. These IGBT combinations are summarised by Table 2.1. Note that the reverse voltage rating of each device only needs to be half of the total DC bus voltage.

Table 2.1: Switch states of a three-level NPC inverter

Switch States				
S_1	S_2	S_3	S_4	Output Voltage
ON	ON	OFF	OFF	$+½V_S$
OFF	ON	ON	OFF	0
OFF	OFF	ON	ON	$-½V_S$

Figure 2.2 shows the neutral point as being created by using a capacitor voltage divider to split the DC bus into two halves. Provided the energy taken from each of the capacitors is equal, the neutral point will remain at the mid-point of the DC supply. In an electric vehicle application, the neutral point is easily obtained by tapping off the mid-point of the battery bank.

2.1.3 d,q Voltage Model

Advanced induction motor control techniques such as field oriented control and direct torque control operate a d,q reference frame. In a balanced three-phase system, the three-phases contain redundant information and can be transformed into an equivalent two-phase system without loss of information [Novotny and Lipo, 1997]. It is therefore useful to have a model of the inverter in this reference frame.

With three states (positive, zero, and negative) for each of the three phases, a total of 3^3 (27) valid IGBT combinations can be produced. Ten out of the 27 possible combinations are listed in Table 2.2. Each IGBT combination is represented as a mnemonic such as +− 0 that represents the voltage produced by each of the three phases. Table 2.2 also lists the corresponding phase-to-phase voltages for each combination. The phase-to-phase voltage applied to the load is either zero (both phases at the same state), half of the DC bus voltage (one phase at zero and the other either positive or negative), or the full DC bus voltage (one phase positive and the other negative).

Table 2.2: Various Three-Level Inverter States

State	V_{AB}	V_{BC}	V_{CA}	V_Q	V_D
---	0	0	0	0	0
000	0	0	0	0	0
+++	0	0	0	0	0
0--	$\frac{1}{2}V_{DC}$	0	$-\frac{1}{2}V_{DC}$	$\frac{1}{3}V_{DC}$	0
+00	$\frac{1}{2}V_{DC}$	0	$-\frac{1}{2}V_{DC}$	$\frac{1}{3}V_{DC}$	0
00-	0	$\frac{1}{2}V_{DC}$	$-\frac{1}{2}V_{DC}$	$\frac{1}{6}V_{DC}$	$\frac{1}{2\sqrt{3}}V_{DC}$
++0	0	$\frac{1}{2}V_{DC}$	$-\frac{1}{2}V_{DC}$	$\frac{1}{6}V_{DC}$	$\frac{1}{2\sqrt{3}}V_{DC}$
+--	V_{DC}	0	$-V_{DC}$	$\frac{2}{3}V_{DC}$	0
+0-	$\frac{1}{2}V_{DC}$	$\frac{1}{2}V_{DC}$	$-V_{DC}$	$\frac{1}{2}V_{DC}$	$\frac{1}{2\sqrt{3}}V_{DC}$
++-	0	V_{DC}	$-V_{DC}$	$\frac{1}{3}V_{DC}$	$\frac{1}{\sqrt{3}}V_{DC}$

The d,q voltage produced by each combination can be calculated. Equation (2-1) shows the transform of phase-to-phase voltages to phase-to-neutral voltages, while equation (2-2) shows the three-phase to d,q transformation. These two sets of equations can be combined to create (2-3), the d,q voltage in terms of the phase-to-phase voltages. Equation (2-3) is only valid for a balanced three-phase system.

$$\begin{aligned} V_A &= \tfrac{1}{3}(V_{AB} - V_{CA}) \\ V_B &= \tfrac{1}{3}(V_{BC} - V_{AB}) \\ V_C &= \tfrac{1}{3}(V_{CA} - V_{BC}) \end{aligned} \tag{2-1}$$

$$\begin{aligned} V_Q &= \tfrac{2}{3}V_A - \tfrac{1}{3}V_B - \tfrac{1}{3}V_C \\ V_D &= \tfrac{1}{\sqrt{3}}V_B - \tfrac{1}{\sqrt{3}}V_C \end{aligned} \tag{2-2}$$

$$\begin{aligned} V_Q &= \tfrac{1}{3}(V_{AB} - V_{CA}) \\ V_D &= \tfrac{1}{\sqrt{3}}\left(\tfrac{-1}{3}V_{AB} + \tfrac{2}{3}V_{BC} - \tfrac{1}{3}V_{CA}\right) \end{aligned} \tag{2-3}$$

Plotting the d,q voltages produced by all 27 combinations results in the d,q vector plot of Figure 2.3. Each point represents a voltage vector in the d,q reference frame that the

three-level inverter can produce. By quickly switching the inverter between these voltage vectors, they can be averaged to produce any voltage vector that lies within the outer hexagon. The technique is called Space Vector Modulation (SVM) and is described in detail in chapter 4.

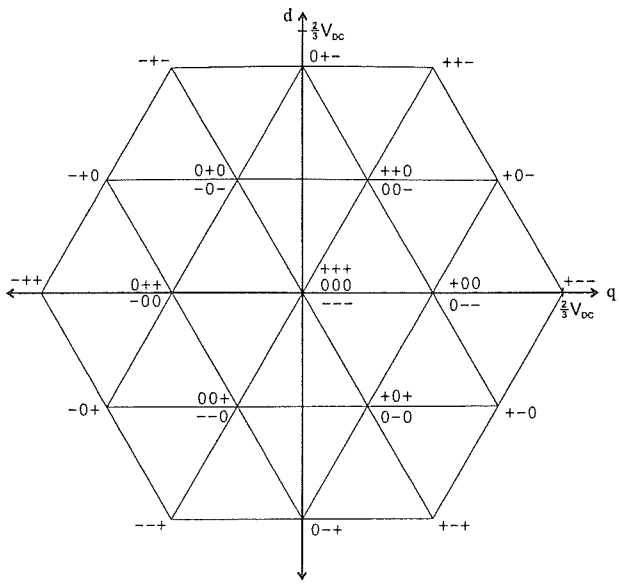


Figure 2.3: Voltage vectors produced by a three-level inverter

2.1.4 Neutral Point Voltage Variation

While there are 27 possible IGBT combinations, there are only 19 unique voltage vectors shown in Figure 2.3 since some of the combinations produce the same voltage vector. There are three different inverter states that will produce the zero voltage vector and two states for each of the six inner voltage vectors of Figure 2.3. Consider the inverter states of `+00` and `0--` highlighted in Table 2.2. These both produce the same phase-to-phase voltage and therefore the same voltage vector.

The difference between each of the inverter states that generate the same voltage vector is in the way the load is connected to the DC bus. Figure 2.4 shows that the 27 inverter states can be divided into five categories.

a) All phases of the load are connected to the same potential, either the positive rail (as shown), neutral point, or the negative rail. As a result no net voltage is applied to the load. This corresponds to the states `+++`, `000`, and `---`.

b) Only the positive and negative rails of the DC bus are used. This corresponds to a two-level inverter and has no effect on the potential of the neutral point. This corresponds to the states +−−, ++−, −+−, −++, −−+, and +−+.

c) In this case the load is connected to the positive rail, neutral point, and negative rail. The affect on the neutral point depends on the load current. This corresponds to the states +0−, 0+−, −+0, −0+, 0−+, and +−0.

d) In this case the load is connected to the upper half of the bus only, causing capacitor C_1 to discharge and C_2 to charge. As a result the voltage of the neutral point starts to rise. This corresponds to the states +00, ++0, 0+0, 0++, 00+, and +0+.

e) This is the opposite to case d). The load is connected to the lower half of the bus only, causing C_2 to discharge and C_1 to charge. Therefore the neutral point voltage starts to decrease. This corresponds to the states 0−−, 00−, −0−, −00, −−0, and 0−0.

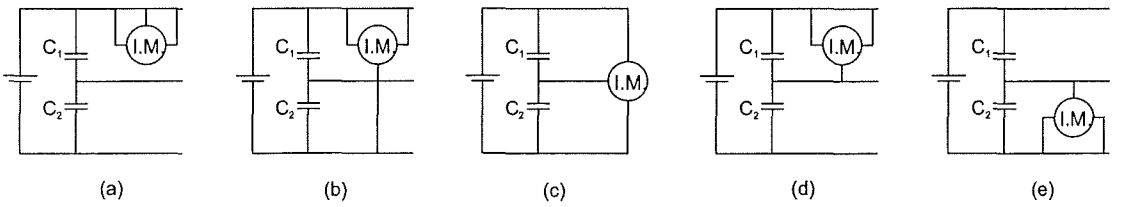


Figure 2.4: Possible Load-Supply Interconnections

This analysis shows that the voltage of the neutral point is affected by the inverter state. The selection of these states must therefore be performed to keep the neutral point at the mid-point of the DC bus. There are two possible states for each of the six inner voltage vectors in Figure 2.3. These correspond to categories d) and e) in Figure 2.4 and can be used to control the neutral point voltage. This adds to the complexity of performing pulse width modulation for a three-level inverter. The problem of neutral point voltage variation is described in more detail in Section 4.2.3.

2.2 Induction Motor Model

For the majority of variable speed induction motor applications such as pumps, fans, and compressors, highly accurate speed control and a fast transient response is not required.

The design of control systems for these applications is based on the 'per phase' T equivalent circuit. This equivalent circuit is derived assuming steady state conditions and is based on an equivalent rotor resistance that is dependant on the induction motor slip speed. This model of the induction model allows rough control over speed, but is not suitable where the transient response and low speed performance is critical.

A much more accurate transient induction motor model can be developed by considering the torque developed by the motor as the cross product of flux and current. Accurate motor control can then be achieved by controlling the spatial position of the flux and current. This is in contrast to the per phase equivalent circuit approach that views torque production as being frequency dependant only, i.e. dependant on the slip.

The full derivation of the transient induction motor model in a d,q reference frame can be found in the book written by Novotny and Lipo [Novotny and Lipo, 1997]. The resulting model is shown as equations (2-7) to (2-17). A number of assumptions have been made in the induction motor model presented.

- A squirrel cage rotor is assumed and therefore no external voltage is applied to the rotor.
- The stator windings are either delta connected or wye connected such that no neutral current flows. This means there is no zero sequence current or voltage.
- The equations are shown for an arbitrary reference frame rotating at a speed of ω. The equations can be converted to the stator (stationary) reference frame by setting $\omega = 0$ or to the rotor reference frame by setting $\omega = \omega_r$.

Table 2.3 defines the symbols that are used by the induction motor model described in this section.

Table 2.3: Induction Motor Model Symbol Definitions

Symbol	Definition
v_{ds} , v_{qs}	Stator voltage (Volts)
i_{ds} , i_{qs}	Stator current (Amps)
λ_{ds} , λ_{qs}	Stator magnetic flux (Webers)
i_{dr} , i_{qr}	Rotor current (Amps)
λ_{dr} , λ_{qr}	Rotor magnetic flux (Webers)
r_s , r_r	Stator / rotor resistance (Ohms)
L_{ls} , L_{lr}	Stator / rotor leakage inductance (Henrys)
L_s , L_r	Stator / rotor self inductance (Henrys)
L_m	Magnetising inductance (Henrys)
P	Number of poles
ω	Rotational speed of the arbitrary reference frame (rad/sec)
ω_r	Rotor electrical speed (rad/sec). This is the rotor mechanical speed multiplied by the number of pole pairs.
T_{em}	Electromagnetic torque produced

Equations (2-4) and (2-5) define the stator and rotor self-inductance as the combination of leakage and magnetising inductance terms. The magnetising inductance term is due to flux that crosses the air gap and links the stator windings and rotor bars. The stator and rotor leakage inductance terms account for the leakage flux. This is flux generated by the stator or rotor that fails to cross the air-gap.

$$L_s = L_{ls} + L_m \tag{2-4}$$

$$L_r = L_{lr} + L_m \tag{2-5}$$

Equation (2-6) defines the stator transient inductance. This is a combination of inductance terms commonly found during the derivation of some equations.

$$L_s' = L_s - \frac{L_m^2}{L_r} \tag{2-6}$$

The induction motor stator equations are shown as (2-7) through (2-10). Equations (2-7) and (2-8) show the applied voltage will cause a change in the stator flux and that some of the applied voltage is lost due to the voltage drop across the stator resistance. These equations show cross coupling exists between the d and q axes. The effect of this cross coupling can be removed by the appropriate selection of the reference frame. The flux on the stator side of the air-gap, (2-9) and (2-10), is the combination of the flux due to the stator current flowing in the stator windings and the component of the flux produced by the rotor that crosses the air-gap.

$$v_{qs} = r_s i_{qs} + \frac{d\lambda_{qs}}{dt} + \omega\lambda_{ds} \tag{2-7}$$

$$v_{ds} = r_s i_{ds} + \frac{d\lambda_{ds}}{dt} + \omega\lambda_{qs} \tag{2-8}$$

$$\lambda_{qs} = L_s i_{qs} + L_m i_{qr} \tag{2-9}$$

$$\lambda_{ds} = L_s i_{ds} + L_m i_{dr} \tag{2-10}$$

The induction motor rotor equations are similar to the stator equations and are shown as (2-11) through (2-14). The rotor voltage equations, (2-11) and (2-12), are dependent on the slip speed relative to the reference frame. The flux on the rotor side of the air-gap, (2-13) and (2-14), is the combination of the flux due to the rotor current flowing in the rotor bars and the component of the flux produced by the stator that crosses the air-gap.

$$0 = r_r i_{qr} + \frac{d\lambda_{qr}}{dt} + (\omega - \omega_r)\lambda_{dr} \tag{2-11}$$

$$0 = r_r i_{dr} + \frac{d\lambda_{dr}}{dt} - (\omega - \omega_r)\lambda_{qr} \tag{2-12}$$

$$\lambda_{qr} = L_r i_{qr} + L_m i_{qs} \tag{2-13}$$

$$\lambda_{dr} = L_r i_{dr} + L_m i_{ds} \tag{2-14}$$

Equations (2-15), (2-16), and (2-17) define three different expressions for the torque produced by the induction motor. These three equations are shown since they are used in the descriptions of the induction motor torque control schemes presented in subsequent sections, although many other expressions for the torque are possible. Equation (2-15) defines the torque as the cross product of rotor flux and stator current, (2-16) defines the

torque as the cross product of rotor flux and stator flux, and (2-17) defines the torque as the cross product of stator flux and stator current.

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}\right) \tag{2-15}$$

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_s' L_r}\left(\lambda_{dr} \lambda_{qs} - \lambda_{qr} \lambda_{ds}\right) \tag{2-16}$$

$$T_{em} = \frac{3}{2}\frac{P}{2}\left(\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}\right) \tag{2-17}$$

Equations (2-7) through (2-17) can be written more compactly by using vector notation. A vector $\overline{f}$ can be defined as shown by equation (2-18), where f is any of the symbols of Table 2.3 that are expressed as d and q components.

$$\overline{f} = \begin{bmatrix} f_q \\ f_d \end{bmatrix} \tag{2-18}$$

The induction motor model presented in this section is used as the basis for the description of the induction motor control schemes that are presented in the following section and in chapter 3.

2.3 Induction Motor Torque Control

It was noted in the previous section that accurate induction motor control can only be achieved by controlling the spatial position of flux and current. This can be compared to a DC motor with separate armature and field windings. The commutator is arranged such that the current in the armature is spatially orientated 90^0 relative to the field. Controlling the current in the field windings controls the field flux while controlling the current in the armature windings controls the torque produced. This independent control of flux and torque is only possible due to the 90^0 separation of the armature current and field flux.

The key to high performance induction motor control is the ability to control the flux and torque independently, similar to a DC motor. This is difficult to achieve for an induction motor since the rotor cannot be controlled independently. The applied stator current

induces the rotor magnetic field that then interacts with the stator current to produce torque.

Numerous induction motor control schemes have been developed to enable independent control of torque and flux. These control schemes can generally be divided into two categories, these being Field Oriented Control (FOC) and Direct Torque Control (DTC). Section 2.3.1 describes Field Oriented Control, which involves the separation of the stator current into two components, one that controls the flux and another that controls the torque. Section 2.3.2 describes Direct Torque Control, which directly controls the inverter in order to apply a voltage that will drive the torque and flux towards the reference values. Section 2.3.3 describes a modification to DTC that reduces the ripple in the torque produced and Section 2.3.4 describes some additional DTC type schemes.

2.3.1 Field Oriented Control

Field oriented control was the first technique developed to allow independent control of induction motor torque and flux. It refers to induction motor operation in a synchronously rotating d,q reference frame that is aligned with one of the motor fluxes, typically the rotor flux [Novotny and Lipo, 1997]. In this mode of operation, control of the torque and flux is decoupled such that the d-axis component of the stator current controls the rotor flux magnitude and the q-axis component controls the torque produced. This was initially difficult to implement due to the complexity of transforming the three phase variables to a rotating d,q reference frame. With the development of suitable low cost microprocessors in the early 1980s, field oriented control become practical to implement in commercial motor drives.

The induction motor model of Section 2.2 can be expressed in the rotor flux reference frame by setting $\omega = \omega_e$, where ω_e is the instantaneous speed of the rotor flux vector $\overline{\lambda_r}$. The phase of the reference frame is aligned such that the rotor flux is entirely in the d-axis, i.e. $\lambda_r^* = \lambda_{dr}$. Since the rotor flux is entirely in the d-axis, $\lambda_{qr} = 0$. Notice that the superscript '*' is used to indicate a reference value.

Since $\lambda_{qr} = 0$, equation (2-12) can be written as (2-19). Combining this with equation (2-14) results in (2-20), an expression for the d-axis stator current in terms of the rotor flux reference. If the value of the flux reference is constant, equation (2-20) can be simplified to (2-21).

$$0 = r_r i_{dr} + \frac{d\lambda_r^*}{dt} \tag{2-19}$$

$$i_{ds} = \frac{1}{L_m}\lambda_r^* + \frac{L_r}{r_r L_m}\frac{d\lambda_r^*}{dt} \tag{2-20}$$

$$i_{ds} = \frac{1}{L_m}\lambda_r^* \tag{2-21}$$

Since $\lambda_{qr} = 0$, equation (2-15) for the torque as a function of rotor flux and stator current can be written as (2-22). This can also be written as equation (2-23), an expression for the q-axis stator current in terms of the torque and flux reference values.

$$T_{em}^* = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\lambda_r^* i_{qs} \tag{2-22}$$

$$i_{qs} = \frac{2}{3}\frac{2}{P}\frac{L_r}{L_m}\frac{T_{em}^*}{\lambda_r^*} \tag{2-23}$$

In the rotor reference frame, equation (2-11) becomes (2-24) and since $\lambda_{qr} = 0$, equation (2-13) can be written as (2-25). Substituting equations (2-21) and (2-25) into (2-24) results in equation (2-26). Equation (2-26) is called the 'slip relation'. For the d-axis of the synchronously rotating reference frame to be aligned with the rotor flux, the 'slip relation' must be maintained.

$$0 = r_r i_{qr} + (\omega_e - \omega_r)\lambda_{dr} \tag{2-24}$$

$$i_{qr} = \frac{-L_m}{L_r} i_{qs} \tag{2-25}$$

$$\omega_e - \omega_r = \frac{r_r}{L_r}\frac{i_{qs}}{i_{ds}} \tag{2-26}$$

To summarise the preceding equations;

1. The d-axis component of the stator current is determined based on the rotor flux reference value using equation (2-21).

2. The q-axis component of the stator current is determined based on the torque and flux reference values using equation (2-23).
3. Based on the calculated d,q stator current, there is a unique slip speed that is determined by the slip relation of equation (2-26). This slip speed must be maintained to achieve correct decoupling of the torque and flux components of the stator current.

A diagram of a field-oriented controller is shown in Figure 2.5. The field oriented controller block shown uses equations (2-21), (2-23) and (2-26) to calculate the stator current and the required slip speed. The slip speed is added to the measured rotor speed to determine the required excitation frequency. A PI controller regulates the voltage applied to the motor to achieve the required stator current. The voltage determined by the PI controller is synthesised using an inverter with some form of pulse width modulation, typically space vector modulation [Holtz, 1994]. A three-level inverter can be used to synthesise the voltage, resulting in less voltage distortion and therefore lower current harmonics.

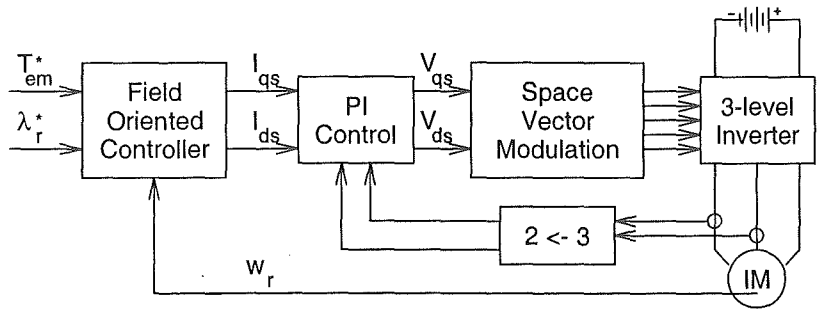


Figure 2.5: Field Oriented Control Block Diagram

During motor operation, the actual rotor resistance and inductance can vary, for example with temperature. The resulting errors between the values used and the actual motor parameters causes an incomplete decoupling between torque and flux. This results in a degraded transient response and steady state errors in the torque and flux produced.

2.3.2 Direct Torque Control

The direct torque control (DTC) scheme was first introduced in the mid 1980s [Takahashi and Noguchi, 1986], [Depenbrock, 1988]. The DTC scheme selects between the inverter's six non-zero voltage vectors and two zero vectors in order to keep the stator flux and torque within a hysteresis band around the flux and torque reference values.

If the induction motor model of Section 2.2 is considered in the stator reference frame ($\omega = 0$), then equations (2-7) and (2-8) can be combined and written in vector form as equation (2-27). If the voltage drop due to the stator resistance is ignored, equation (2-27) can be approximated over a short time period as (2-28). This shows that the applied voltage determines the change in the stator flux vector.

$$\overline{v_s} = r_s \overline{i_s} + \frac{d\overline{\lambda_s}}{dt} \tag{2-27}$$

$$\Delta\overline{\lambda_s} = \overline{v_s} \cdot \Delta t \tag{2-28}$$

Equation (2-16), for the torque produced in terms of the stator and rotor flux, can be written as (2-29). This shows the torque produced is dependent on the stator flux magnitude, rotor flux magnitude, and the phase angle between the stator and rotor flux vectors. If the stator flux and rotor flux magnitudes are fixed, controlling the phase angle (α) controls the torque produced.

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_s' L_r}\left|\overline{\lambda_r}\right| \cdot \left|\overline{\lambda_s}\right| \cdot \sin\alpha \tag{2-29}$$

If the applied voltage is in the direction of the stator flux vector, its magnitude will be increased. If the applied voltage is perpendicular to the stator flux vector, the stator flux vector will be rotated and the torque produced will increase or decrease depending on the direction of rotation.

Figure 2.6 shows the operation of direct torque control for a two-level inverter. The vector space is divided into six sectors such that the six voltage vectors that the inverter can produce reside at the centre of each sector. Also shown is the stator flux vector, $\overline{\lambda_s}$, and the rotor flux vector, $\overline{\lambda_r}$. Shown at the end of the stator flux vector are four vectors

labelled $\Delta\overline{\lambda_s}$. These four vectors are the change in stator flux corresponding to applying the voltage vectors $\overline{V_2}$, $\overline{V_3}$, $\overline{V_5}$, and $\overline{V_6}$. Each of these four voltage vectors has a different effect on the stator flux vector.

1. $\overline{V_2}$ will increase the magnitude of the stator flux vector and rotate it away from the rotor flux vector, increasing the torque produced
2. $\overline{V_3}$ will decrease the magnitude of the stator flux vector and rotate it away from the rotor flux vector, increasing the torque produced
3. $\overline{V_5}$ will decrease the magnitude of the stator flux vector and rotate it towards the rotor flux vector, decreasing the torque produced
4. $\overline{V_6}$ will increase the magnitude of the stator flux vector and rotate it towards the rotor flux vector, decreasing the torque produced.

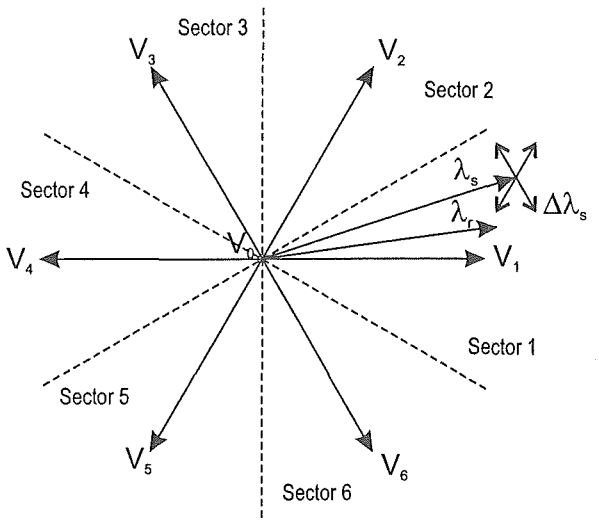


Figure 2.6: Direct torque control for a two-level inverter

If the zero voltage vector ($\overline{V_0}$) is applied, equation (2-28) shows the stator flux vector will not change. In practise, the magnitude will decrease slightly due to the stator resistance and the torque will decrease as the rotor flux vector rotates forward due to the rotating rotor.

The effect of each of the voltage vectors depends on which of the six sectors the stator flux vector resides in. A table can be constructed that specifies the effect on the torque and flux magnitudes for each of the voltage vectors for each of the six sectors. This is

called the optimal vector selection table and is shown as Table 2.4 for the case of a two level inverter.

Table 2.4: Optimal vector selection table for a two-level inverter

		Sector					
$\Delta\lambda$	ΔT_{em}	1	2	3	4	5	6
↑	↑	V_2	V_3	V_4	V_5	V_6	V_1
	0	V_0	V_0	V_0	V_0	V_0	V_0
	↓	V_6	V_1	V_2	V_3	V_4	V_5
↓	↑	V_3	V_4	V_5	V_6	V_1	V_2
	0	V_0	V_0	V_0	V_0	V_0	V_0
	↓	V_5	V_6	V_1	V_2	V_3	V_4

The voltage vectors of Table 2.4 are selected in order to drive the stator flux magnitude and torque to the reference values. This requires an estimate of the instantaneous stator flux vector and the torque produced by the motor.

The stator flux vector can be estimated using the stator voltage equations, (2-7) and (2-8), which in the stator reference frame ($\omega = 0$) can be written as equation (2-30).

$$\bar{\lambda}_s = \int (\bar{v}_s - r_s \bar{i}_s) dt \qquad (2\text{-}30)$$

Once the stator flux is estimated, the torque produced can be calculated using the torque equation of (2-17).

Hysteresis comparators are used to select the vector from Table 2.4 that will drive the error between the estimated parameters and the reference values to zero. A three-level hysteresis comparator is used for the torque error and a two-level hysteresis comparator for the flux error. Figure 2.7 shows the operation of a three-level hysteresis comparator. The comparator output is '1' if the torque error exceeds ΔT_{em}. A voltage vector is then selected to decrease the torque. As the torque error decrease below '0', the zero voltage vector is selected. If the torque error drops below $-\Delta T_{em}$, a vector is selected to increase the torque. The average switching frequency is determined by the width of the hysteresis bands [Casadei, Grandi et al., 1994] and also by the length of time it takes to perform the calculations and select the voltage vector.

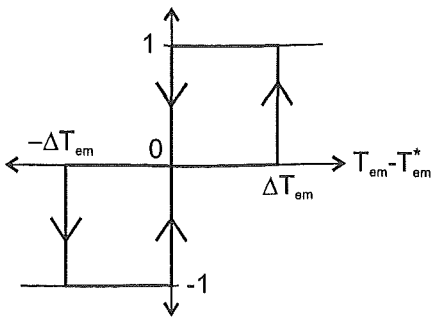


Figure 2.7: Operation of a three-level hysteresis comparator

Using multiple two-level inverters [Takahashi and Ohmori, 1989] or a single multi-level inverter [Martins, Carvalho et al., 1998] increases the number of voltage vectors available for selection. This allows better control of the applied voltage and therefore the stator flux vector. This reduces the torque ripple, flux ripple and current distortion. The extra voltage vectors allow the vector selection table, Table 2.4, to be expanded [Takahashi and Ohmori, 1989]. The vector space of Figure 2.6 can be divided into twelve sectors instead of six and the torque change can be dividing into five levels, a large decrease, small decrease, no change, small increase, and large increase. The extra torque changes are selecting by replacing the three-level torque hysteresis comparator with a five-level comparator.

Figure 2.8 shows a block diagram of the direct torque control scheme. The estimated stator flux magnitude and torque are compared to the reference values. A single voltage vector is then selected that will drive the error in both parameters to zero.

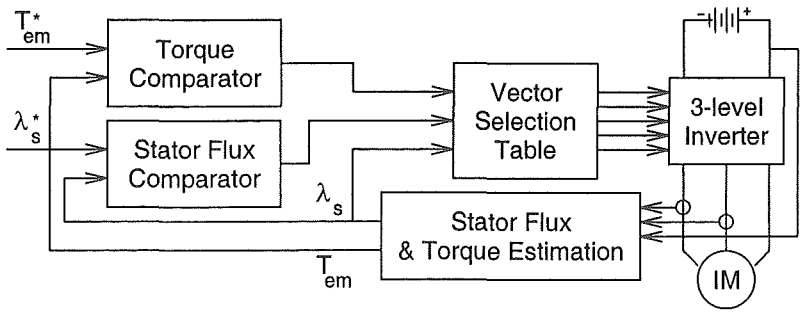


Figure 2.8: Direct Torque Control Block Diagram

Direct torque control enables rapid changes in the torque and flux compared to FOC due to the lack of a PI current controller and no encoder is required to measure the rotor speed. The disadvantage of DTC is the high torque and flux ripple, and high current distortion that it produces. It also has poor performance at low speed due to problems with the stator flux estimation.

2.3.3 Minimal Torque Ripple DTC

As an improvement to the original DTC scheme described in Section 2.3.2, the switching instant between the non-zero and zero voltage vectors can be calculated to minimise torque ripple [Kang and Sul, 1999].

During steady state operation, DTC will switch between a non-zero voltage vector and the zero voltage vector. Typically a non-zero vector is selected to increase the torque while maintaining control of the stator flux magnitude. Once the upper limit of the torque hysteresis band is reached, the zero voltage vector is applied and the torque slowly decreases as shown in Figure 2.9.

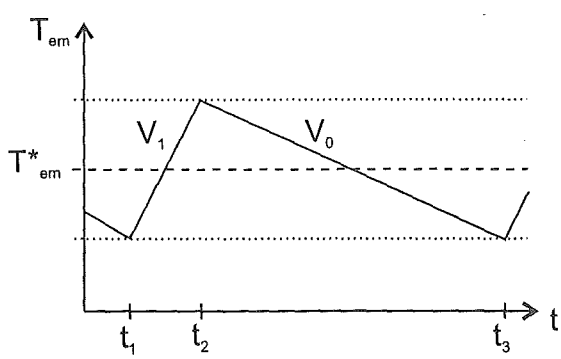


Figure 2.9: Effect of voltage vectors on the torque

By calculating the optimal time to switch between the non-zero and zero voltage vectors, the torque ripple can be minimised. This calculation is achieved by using the induction motor model to calculate the rate of torque increase due to the selected non-zero voltage vector and the rate of torque decrease due to the zero voltage vector. The time to switch between the two vectors can be calculated such that the ripple in the torque over a fixed switching period can be minimised. Figure 2.10 shows the basic direct torque control

block diagram with an added torque ripple minimisation block. This block applies the selected voltage vector from t_1 to t_2 and the zero vector from t_2 to t_3.

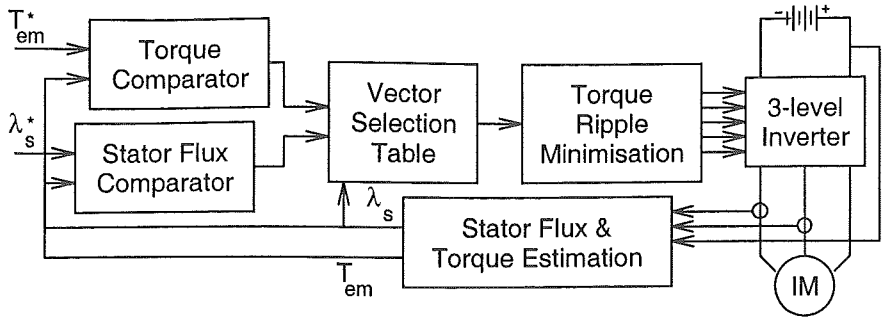


Figure 2.10: DTC with Minimal Torque Ripple Block Diagram

The ratio between the non-zero and zero voltage vectors is calculated to minimise torque ripple without concern for the effect on stator flux. The result is a much greater variation in stator flux compared to standard DTC. In standard DTC, a new vector would be selected as soon as the stator flux leaves the hysteresis band. In this case a vector is not selected to change the stator flux until the next cycle.

The minimal torque ripple DTC was originally published [Kang and Sul, 1999] as a technique to directly control a two-level inverter. When using a three-level inverter, the vector space can be divided into 12 sectors instead of 6, like for three-level classical DTC. This presents a problem for minimal torque ripple DTC as there are two types of voltage vector available for selection, the smaller (inner) voltage vectors and the larger (outer) voltage vectors. Classical DTC uses a multi-band hysteresis comparator to select which type of vector to use, if the error exceeds the inner band then the larger voltage vectors are used. This is not possible for minimal torque ripple DTC.

One solution would be to repeat the calculations using both the selected inner vector and the selected outer vector. If no solution is found using the inner voltage vector, it means that the voltage is insufficient to produce the required torque and the calculations should be repeated using the larger voltage vector. Unfortunately this doubles the number of calculations that must be performed. To keep the three-level extension of the minimal torque ripple DTC scheme simple, only the 12 outer voltage vectors are used. This does

not increase the calculation time of the original scheme, but ignoring the six inner voltage vectors results in increased torque ripple when operating at low speed and torque.

2.3.4 Other Torque Control Schemes

Numerous other control schemes were investigated during a review of the available literature. One scheme investigated is called Direct Torque Control using Space Vector Modulation and is described in chapter 3. Two other schemes called DTC using discrete SVM and vectorial torque control were also considered.

DTC using discrete SVM [Casadei, Serra et al., 2000] uses a fixed sample period that is divided into a number of smaller intervals with a different voltage vector being applied during each interval. By dividing the sample period into three intervals, 19 unique voltage vectors can be synthesised by the inverter with up to three inverter states being used to synthesise each vector. The DTC vector selection table can then be extended to use the extra voltage vectors, similar to a three-level inverter. Applying this technique to a three-level vector would result in even more vectors available for selection and greatly increase the complexity of the vector selection table.

Vectorial Torque Control [Attaianese, Nardt et al., 1999] predicts the torque and flux resulting from each of the inverter's voltage vectors and chooses a modulation pattern to achieve the reference torque and flux values. The torque produced by each of the voltage vectors that the inverter can produce is calculated. Two vectors are then selected, one to increase the torque and another to decrease it. The ratio between the two vectors is calculated to such that they average to the reference torque. The combination of vectors that results in minimum deviation of the stator flux from the reference flux is selected. This process would become computationally intensive as the number of voltage vectors increased, as is the case for a three-level inverter.

2.4 Summary

This chapter has introduced the three-level neutral point clamped inverter that is used for this project. It showed that a three-level inverter is able to produce an output voltage with less distortion for the same device switching frequency compared to a two-level inverter.

A voltage model of the inverter in a d,q reference frame was developed to assist in the explanation of the various torque control schemes that also operate in the d,q reference frame. It was also shown that the different inverter states have different effects on the neutral point voltage. Space vector modulation for a three-level inverter is described in detail in chapter 4.

Section 2.2 presented a model of the induction motor in an arbitrary rotating d,q reference frame. This model considers the torque produced as being due to the interaction of magnetic fields and currents within the induction motor and is essential to the development of high performance induction motor control schemes.

Section 2.3 introduced the two basic categories of induction motor torque control schemes. These were field oriented control and direct torque control. A few schemes to improvement the basic direct torque control technique were also presented. Computer simulations are performed in chapter 5 for the three torque control schemes described in detail in this chapter and the fourth scheme that is described in the following chapter.

3. Direct Torque Control

Chapter 2 provided an overview of field oriented control and direct torque control as well as various modifications to the classical direct torque control scheme. A control scheme called direct torque control using space vector modulation is used for this project as it has some advantages. This control scheme is described in detail in this chapter. Section 3.1 provides an overview of direct torque control using space vector modulation and Section 3.2 provides a detailed description of the implementation.

3.1 Overview of DTC using SVM

Direct Torque Control using Space Vector Modulation for an induction motor was first published in 1991 [Habetler, Profumo et al., 1991]. Using only current and voltage measurements, it is possible to estimate the instantaneous stator flux and torque output, similar to classical DTC. An induction motor model is then used to predict the voltage required to drive the torque and flux to the reference values within a fixed time period. The required voltage is synthesised using space vector modulation. If the inverter is not capable of generating the required voltage, a voltage vector that will drive the torque and flux towards the reference values is used. A block diagram of this configuration is shown in Figure 3.1.

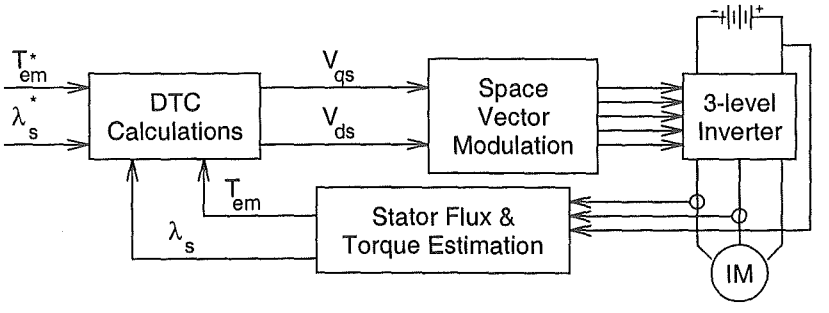


Figure 3.1: DTC using Space Vector Modulation Block Diagram

3.2 Implementation of DTC using SVM

This section describes the implementation of direct torque control using space vector modulation in detail. A number of advances on the implementation originally published [Habetler, Profumo et al., 1991] are possible. These are primarily due to the more sophisticated power electronic and controller hardware that are being used. These advances are:

- An IGBT inverter switching at 10kHz is used instead of a BJT inverter switching at 2kHz.
- A three-level inverter is used instead of a two-level inverter. This enabled a higher quality stator voltage waveform to be synthesised.
- Due to a higher inverter switching frequency and more sophisticated controller hardware platform, the controller update rate has been reduced from 250μs to 100μs.
- The stator flux estimation is performed digitally rather than use analogue integrators.

In the stator d,q reference frame, the induction motor model of Section 2.2 can be represented by the equivalent circuit of Figure 3.2. This views the induction motor as the series combination of a stator resistance (r_s), transient inductance (L_s'), and a back EMF ($\overline{E}$) due to the rotating magnetic field of the rotor.

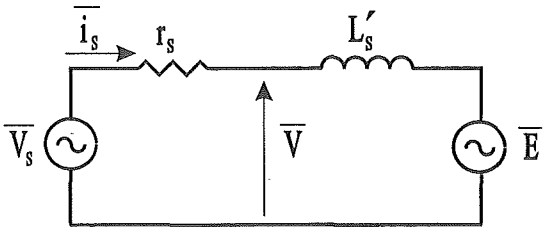


Figure 3.2: Equivalent circuit in the stator d,q reference frame

The equivalent circuit of the induction motor in Figure 3.2 is used in the explanation of DTC using space vector modulation that is presented in the following sections. Section 3.2.1 describes the stator flux estimation and Section 3.2.2 describes the calculation of the

back EMF. The voltage prediction calculations are described in Section 3.2.3. Due to limitations of the inverter, there is a maximum current and voltage the inverter can supply. Section 3.2.4 describes how current limiting is implemented and Section 3.2.5 describes what happens when the inverter cannot generate the required voltage.

3.2.1 Stator Flux Estimation

There are two possible techniques for the estimation of the stator flux vector. These are termed the voltage model stator flux observer and the current model stator flux observer. These are described in Sections 3.2.1.1 and 3.2.1.2 respectively.

3.2.1.1 Voltage Model Stator Flux Observer

The voltage model stator flux observer is developed from the induction motor stator equations in the stator d,q reference frame. This was described in Section 2.3.2 for classical DTC and is presented here again as equation (3-1).

$$\bar{\lambda}_s = \int (\bar{v}_s - r_s \bar{i}_s) dt \qquad (3\text{-}1)$$

There are two problems with performing stator flux estimation using equation (3-1) [Jansen and Lorenz, 1994].

1. At low speed the pure integration of equation (3-1) is sensitive to drift and offset errors. This is an inherent problem with the integration of low frequency signals.
2. Equation (3-1) is highly sensitive to stator resistance detuning at low speed. At high speed the voltage drop due to the stator resistance ($r_s \bar{i}_s$) becomes insignificant compared to the voltage ($\bar{v}_s$) that is being applied to counter the back EMF. At low speed the stator resistance voltage drop becomes significant and errors result due to the variation of the actual stator resistance with temperature.

The advantage of estimating stator flux using equation (3-1) is that no rotor speed measurement is required. It only requires knowledge of the applied voltage and stator current, which can be easily calculated and measured.

3.2.1.2 Current Model Stator Flux Observer

The current model stator flux observer is achieved using the induction motor model of Section 2.2 in a reference frame aligned to the rotor, i.e. $\omega = \omega_r$. In the following analysis, the parameters are assumed to be in the stator reference frame if no superscript is used and in the rotor reference frame if an 'r' superscript is used.

In the rotor reference frame, equations (2-11) and (2-12) for the rotor voltage can be written using vector notation as equation (3-2). Similarly, equations (2-13) and (2-14) for the rotor flux can be written as equation (3-3). Combining (3-2) and (3-3) results in (3-4), an expression for the rotor flux in terms of stator current and motor parameters.

$$0 = r_r \overline{i_r^r} + \frac{d\overline{\lambda_r^r}}{dt} \tag{3-2}$$

$$\overline{i_r^r} = \frac{\overline{\lambda_r^r} - L_m \overline{i_s^r}}{L_r} \tag{3-3}$$

$$\overline{\lambda_r^r} = \int \frac{r_r}{L_r}\left(L_m \overline{i_s^r} - \overline{\lambda_r^r}\right)dt \cdot \tag{3-4}$$

Equation (3-4) requires the stator current in the rotor reference frame. This is achieved using the rotation transformation of equation (3-5), where θ is the rotor position.

$$\begin{bmatrix} i_{qs}^r \\ i_{ds}^r \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} \tag{3-5}$$

Once the rotor flux has been determined using equation (3-4), it is transformed back to the stationary reference frame using the inverse rotation transformation of equation (3-6).

$$\begin{bmatrix} \lambda_{qr} \\ \lambda_{dr} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} \lambda_{qr}^r \\ \lambda_{dr}^r \end{bmatrix} \tag{3-6}$$

Equations (2-6), (2-9), (2-10), (2-13), and (2-14) can be combined to obtain an expression, (3-7), for the stator flux in terms of rotor flux and stator current.

$$\overline{\lambda_s} = \frac{L_m}{L_r}\overline{\lambda_r} + L_s' \overline{i_s} \tag{3-7}$$

In summary, the process for calculating the stator flux is:

1. Transform the stator currents to the rotor reference frame using (3-5)
2. Calculate the rotor flux in the rotor reference frame using (3-4)
3. Transform the rotor flux into the stator reference frame using (3-6)
4. Calculate the stator flux from the rotor flux and stator current using (3-7)

The advantage of the current model observer is its accuracy at low speed, which is where the voltage model observer has problems. Its disadvantage is the sensitivity to errors in the rotor resistance and magnetising inductance parameters when the induction motor is operating near rated slip [Jansen and Lorenz, 1994]. It also has the disadvantage of requiring an incremental encoder to measure the rotor position.

3.2.1.3 Combination of Voltage and Current models

The voltage model stator flux observer is sensitive to stator resistance detuning at low speed while the current model observer is sensitive to rotor resistance and magnetising inductance detuning when operating near rated slip. There have been various attempts to combine the two observers to reduce the errors due to parameter detuning.

One technique published involves using a PI controller to correct the voltage model at low speed [Lascu, Boldea et al., 1998]. A compensating factor is added to the voltage model observer to correct for errors due to the stator resistance detuning and the pure integrator. A PI controller regulates the compensating factor based on the output of the current model observer. The gains of the PI controller are set so that the current model dominates at low speed while the voltage model dominates at high speed. This configuration was attempted but appeared to be unstable when used with direct torque control, which has a significantly faster response than the PI controller. The estimated flux tended to oscillate between the outputs of each of the two observers without a smooth transition.

Another technique is a self-tuning configuration where the current model observer tunes the stator resistance at low speed and the voltage model observer tunes the rotor resistance at high speed [Habetler, Profumo et al., 1998]. At low speed the current model observer is used to estimate the stator flux while at the same time tuning the stator resistance used by the voltage model. When the speed increases above a set point, the voltage model is used to estimate the stator flux while at the same time tuning the rotor

resistance. This self-tuning technique improves the accuracy of both observers and ensures there is no step change in the stator flux estimate during the transition between the two models.

For this project, both types of stator flux observer where implemented and evaluated independently. A configurable hysteresis transition point was included, but the self-tuning to enable a smooth crossover was not implemented due to time constraints.

3.2.2 Back EMF Estimation

The voltage behind the transient inductance, $\overline{E}$, in Figure 3.2 can be estimated using the stator flux and current.

Equation (3-8) can be written from the equivalent circuit of Figure 3.2. This can be combined with (3-1), the stator flux equation, to obtain (3-9). If $\overline{E}$ is assumed to be sinusoidal then equation (3-9) can be approximated as (3-10), which removes the derivative calculation.

$$\overline{E} = \overline{v_s} - r_s \overline{i_s} - \frac{d}{dt}\left(L_s' \overline{i_s}\right) \tag{3-8}$$

$$\overline{E} = \frac{d}{dt}\left(\overline{\lambda_s} - L_s' \overline{i_s}\right) \tag{3-9}$$

$$\overline{E} = j\omega_e \left(\overline{\lambda_s} - L_s' \overline{i_s}\right) \tag{3-10}$$

The applied excitation frequency required by (3-10) can be estimated from the rotational speed of the stator flux vector [Xue, Xu et al., 1990]. Equation (3-11) shows an expression for the stator flux angle. Taking the derivative as shown by equation (3-12), results in an expression for the rotational speed, (3-13). Replacing the flux derivatives with the stator voltage equations, (2-7) and (2-8), results in (3-14).

$$\theta_s = \tan^{-1}\left(\frac{\lambda_{ds}}{\lambda_{qs}}\right) \tag{3-11}$$

$$\omega_e = \frac{d\theta_s}{dt} = \frac{d}{dt}\tan^{-1}\left(\frac{\lambda_{ds}}{\lambda_{qs}}\right) \tag{3-12}$$

$$\omega_e = \frac{\dot{\lambda}_{ds}\lambda_{qs} - \dot{\lambda}_{qs}\lambda_{ds}}{\lambda_{qs}^2 + \lambda_{ds}^2} \tag{3-13}$$

$$\omega_e = \frac{(v_{ds} - r_s i_{ds})\lambda_{qs} - (v_{qs} - r_s i_{qs})\lambda_{ds}}{\lambda_{qs}^2 + \lambda_{ds}^2} \tag{3-14}$$

3.2.3 Stator Voltage Prediction

The change in stator current over a short time period (t_s) can be approximated as linear, and from the equivalent circuit of Figure 3.2, equation (3-15) can be obtained.

$$\Delta\bar{i}_s = \frac{\bar{V} - \bar{E}}{L_S'} t_s \tag{3-15}$$

The voltage $\bar{V}$ (as defined by (3-16)) is the applied voltage ignoring the voltage drop due to the stator resistance. Temporarily ignoring the stator resistance by defining $\bar{V}$ simplifies the calculations of the required stator voltage.

$$\bar{V} = \bar{v}_s - r_s \bar{i}_s \tag{3-16}$$

Equation (2-17) of the induction motor model showed that the torque produced can be considered as the cross product between stator flux and stator current. This can be used to calculate the instantaneous torque produced by the motor and to develop equation (3-17), which shows the change in torque is the cross product of stator flux and the change in the stator current.

$$\Delta T_{em} = \frac{3}{2}\frac{P}{2}\left(\bar{\lambda}_s \times \Delta\bar{i}_s\right) \tag{3-17}$$

Substituting (3-15) into (3-17) results in (3-18), which can be expanded to (3-19).

$$\Delta T_{em} = \frac{3}{2}\frac{P}{2}\frac{t_s}{L_s'}\left(\bar{\lambda}_s \times \left(\bar{V} - \bar{E}\right)\right) \tag{3-18}$$

$$\Delta T_{em} = \frac{3}{2}\frac{P}{2}\frac{t_s}{L_s'}\left(\left(-\lambda_{ds}E_q + \lambda_{qs}E_d\right) + \left(\lambda_{ds}V_q - \lambda_{qs}V_d\right)\right) \tag{3-19}$$

This shows the change in torque due to an applied voltage $\bar{V}$ in terms of stator flux and back EMF. Equation (3-19) is used to calculate the voltage required to change the torque from the instantaneous value to the reference value within the time period t_s.

The applied voltage $\overline{V}$ will also have an effect on the stator flux vector. The applied voltage must drive the magnitude of the stator flux vector to the reference value by the end of the switching period t_s. This requirement is expressed as (3-20) and can be expanded as (3-21).

$$\lambda_s^* = \left|\overline{V}t_s + \overline{\lambda_s}\right| \tag{3-20}$$

$$\left(\lambda_s^*\right)^2 = \left(V_q t_s + \lambda_{qs}\right)^2 + \left(V_d t_s + \lambda_{ds}\right)^2 \tag{3-21}$$

Equations (3-19) and (3-21) are a linear system of two equations with two unknowns, these being the required voltages V_q and V_d. These two equations can be solved to find the voltage that needs to be applied to cause the required changes in torque and flux. Equation (3-19) can be split into two equations, (3-22) and (3-23), by defining the constant K_e.

$$K_e = \frac{4\Delta T_{em} L_s'}{3Pt_s} + \left(\lambda_{ds} E_q - \lambda_{qs} E_d\right) \tag{3-22}$$

$$V_q = \frac{K_e + \lambda_{qs} V_d}{\lambda_{ds}} \tag{3-23}$$

Substituting equation (3-23) into equation (3-21) results in (3-24), a quadratic equation with V_d as the unknown. Solving this quadratic equation results in two values for V_d. Equation (3-23) is then used to calculate the corresponding V_q for both values. The voltage vector with the lowest magnitude is then selected.

$$\begin{aligned}&\left(t_s^2 + \frac{\lambda_{qs}^2}{\lambda_{ds}^2} t_s^2\right) V_d^2 + \left(\frac{2K_e \lambda_{qs} t_s^2}{\lambda_{ds}^2} + 2\lambda_{ds} t_s + \frac{2\lambda_{qs}^2 t_s}{\lambda_{ds}}\right) V_d \\ &+ \left(\frac{t_s^2 K_e^2}{\lambda_{ds}^2} + \frac{2\lambda_{qs} K_e t_s}{\lambda_{ds}} + \lambda_{ds}^2 + \lambda_{qs}^2 - \left(\lambda_s^*\right)^2\right) = 0\end{aligned} \tag{3-24}$$

The voltage vector $\overline{V}$ has therefore been calculated. The stator voltage drop across r_s needs to be added to this to determine the required stator voltage, $\overline{V_s}$, as shown by equation (3-25). The voltage drop is calculated using the current measured from the present cycle and assuming that the change in current over the short time period will have a minimal effect on the voltage drop.

$$\overline{v_s} = \overline{V} + r_s \overline{i_s} \tag{3-25}$$

3.2.4 Current Limiting

The preceding section calculated the voltage vector $\overline{V_s}$ to be applied to the induction motor. The change in current due to the voltage that is about to be applied can be estimated using equation (3-15). If the calculated voltage would cause the current to increase above the current limit of the inverter, the applied voltage can be changed to avoid this situation.

Equation (3-15) shows there is no change in current if the voltage $\overline{V}$ matches the back EMF, $\overline{E}$. The applied stator voltage needed to meet this requirement can be calculated using (3-26).

$$\overline{v_s} = r_s \overline{i_s} + \overline{E} \tag{3-26}$$

If required, a more comprehensive calculation could be performed to calculate the applied voltage based on the available current increase before the inverter current limit is reached.

3.2.5 Transient Conditions

The available DC bus voltage limits the maximum voltage that the inverter can synthesise. It is possible that the voltage calculated in Section 3.2.3 is larger than this maximum voltage. This is typically due to rapid changes in the torque and flux references, since large voltages are required to drive the motor torque and flux to the new values within the short time period, t_s.

Two techniques have been published to handle this condition. The original technique attempted to drive either the torque or flux to the new reference value while maintaining control of the other parameter [Habetler, Profumo et al., 1991].

1. It is first assumed that a transient change in torque has occurred since this is the most frequent and important case. Two voltage vectors from Table 2.4 are selected that increase and decrease the flux while driving the torque towards the

new reference value. The dwell times of the two vectors are calculated in order to maintain the flux at the reference value.

2. If the calculation of the dwell times to maintain the flux reference in the previous step fails, it is assumed that a transient change in the flux reference has occurred. The two appropriate voltage vectors are selected that will drive the flux to the reference value and their dwell times are calculated to maintain control over torque.
3. If the calculation of the dwell times failed for both steps, it is assumed that both the torque and flux references have changed. A single vector is selected from Table 2.4 that will drive both the torque and flux towards the reference values and that vector applied for the complete cycle.

This technique for handling the transient conditions is computationally intensive to implement. The same authors later published a simplified technique [Griva, Habetler et al., 1995] that significantly reduces the computational requirements by assuming a torque transient is the only important case. During transient conditions, the zero voltage vector is not used in order to drive the torque to the reference value as quickly as possible. This corresponds to the case of a voltage vector following a trajectory along the outer hexagon of Figure 2.3. A new voltage vector is calculated that lies on this outer hexagon but still retains the same angle as the original voltage vector. This new voltage will drive the torque towards the reference value but does not maintain control over flux.

Figure 3.3 and Figure 3.4 show a comparison between using the full transient calculations [Habetler, Profumo et al., 1991] and simply choosing one vector and holding it for the entire switching cycle (effectively performing step 3 only). Chapter 5 provides more details on how the simulations were performed. The simulations shown here are based on a two-level inverter and one of the induction motors from the 'Mark II' electric vehicle. For the period of 0.041 to 0.043 seconds, the stator flux reference was reduced from 0.16wb to 0.1wb. For the period of 0.045 to 0.047, the torque reference was reduced from 20Nm to 10Nm.

Figure 3.3 shows better control over torque during step changes in the stator flux reference. Step changes in the stator flux reference do not normally occur in practice. The stator flux reference is normally fixed, or changed continuously with speed in the

field-weakening region of operation. Figure 3.4 shows only a negligible change in stator flux during a torque transient. Due to the 10kHz switching frequency, a single voltage vector is only held for 100μs until a new vector is selected. Also, the transient conditions only last a few cycles until the new torque reference is reached and full control of torque and flux is restored. These simulations show it is feasible to hold the same vector for the complete switching cycle under transient conditions.

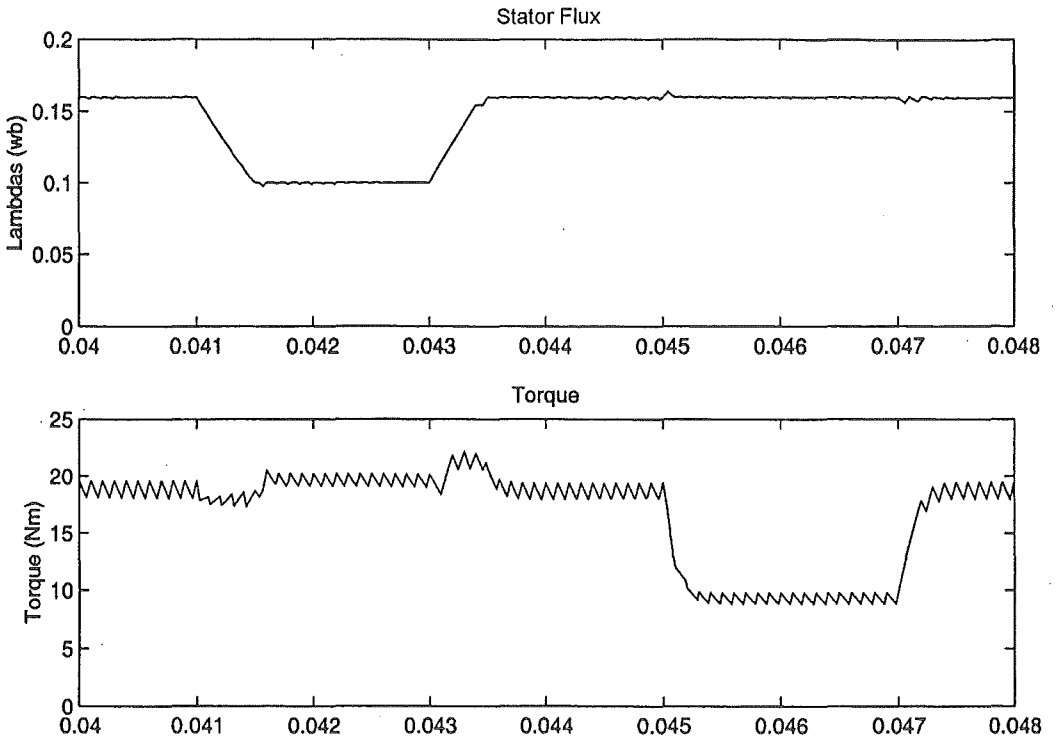


Figure 3.3: Transient Response - Full transient calculations

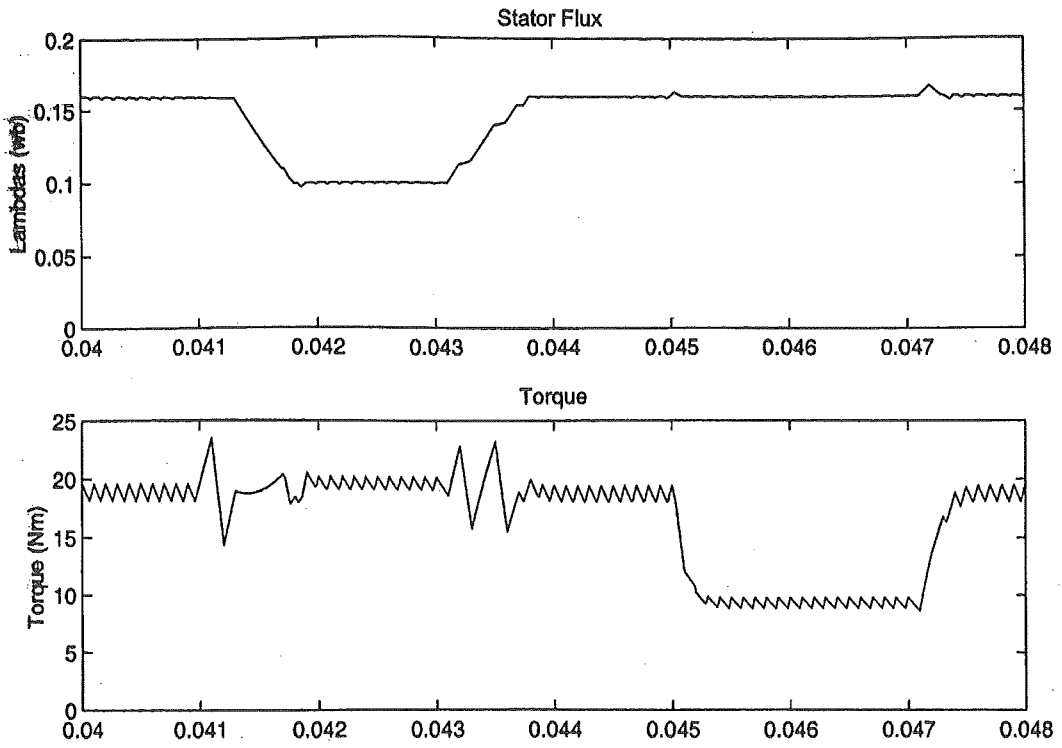


Figure 3.4: Transient Response - Using only a single vector

A rate limit can also be added to the torque reference signal. This reduces the torque change required in each switching cycle and therefore the voltage that is required. The maximum torque change rate is set so that the required voltage can normally be achieved without resorting to holding a single vector and therefore losing accurate flux control capability.

3.2.6 Implementation Summary of DTC using SVM

Sections 3.2.1 to 3.2.5 described the implementation of direct torque control using space vector modulation that is used for this project. The steps required to perform DTC using SVM are summarised as follows:

1. Estimate the stator flux using either the voltage model (equation (3-1)) or the current model (equations (3-4) to (3-7))
2. Estimate the instantaneous motor torque using equation (2-17)
3. Estimate the stator frequency using equation (3-14) and the back EMF using equation (3-10).
4. Calculate the constant K_e using equation (3-22)

5. Solve equation (3-24) to obtain two values for V_d
6. Calculate V_q for both values of V_d using equation (3-23)
7. Select the voltage vector with the lowest magnitude from the two calculated vectors.
8. Add on the effect of the voltage drop due to the stator resistance by using equation (3-25)
9. Recalculate the voltage using equation (3-26) if the inverter current limit would be exceeded
10. Check if the voltage can be synthesised using space vector modulation
    a. If it **can** be synthesised, use the space vector modulation technique in chapter 4 to synthesise the calculated voltage.
    b. If it **cannot** be synthesised, choose the voltage vector from Table 2.4 that will drive the torque and flux towards the reference values and apply that vector for the next switching cycle.

3.3 Summary

Direct torque control using space vector modulation combines the best features of field oriented control and classical direct torque control. It has the low torque ripple, flux ripple, and current distortion characteristic of the space vector modulator that is commonly used for field oriented control. Due to the lack of a PI current controller, it has the fast transient response that is characteristic of direct torque control. No encoder is required if the voltage model stator flux observer is used, but this results in poor low speed performance. Chapter 5 presents simulation results that show a comparison between direct torque control using space vector modulation and the various control schemes described in chapter 2.

The following chapter describes the space vector modulation technique that is used to synthesise the voltage calculated by the direct torque control scheme described in this chapter.

4. Space Vector Modulation

An introduction to the three-level NPC inverter used for this project and its d,q voltage model was presented in chapter 2. This chapter describes a particular pulse width modulation technique called space vector modulation (SVM). SVM is the synthesis of a desired output voltage using the voltage vectors that an inverter is capable of producing. For this project, SVM is used to synthesise the voltage calculated by the direct torque controller described in the previous chapter.

An overview of SVM is provided in Section 4.1. Section 4.2 provides a detailed description of how the voltage vectors, their order, and their dwell times are determined for each switching cycle. In a practical implementation, a short dead-time is needed between IGBT commutations. This introduces an error between the desired output voltage and the actual inverter output voltage. The effect of this dead-time is examined in detail in Section 4.4.

4.1 Space Vector Modulation Overview

A number of existing pulse width modulation (PWM) schemes for two-level inverters have been adapted for three-level operation and new schemes specific to multi-level inverters have been developed [Suh, Sinha et al., 1998]. These modulation schemes include space vector modulation [Lee, Suh et al., 1996], sigma-delta modulation [Manjrekar and Venkataramanan, 1996], switching frequency optimal PWM [Steinke, 1989], and sub-harmonic (sinusoidal) PWM [Carrara, Gardella et al., 1992].

SVM was chosen as the modulation scheme for this project based on its high performance. SVM switches between the voltage vectors that the inverter can produce in order to obtain an average over a fixed switching cycle that is equivalent to a specified reference vector [van der Broeck, Skudelny et al., 1988]. This is achieved by mathematical decomposition of the reference voltage vector into the available inverter voltage vectors. Any reference vector that lies inside the outer hexagon of Figure 2.3 can be approximated using this technique.

SVM achieves a high utilisation of the available DC bus voltage by producing high fundamental output voltage. This is due to space vector modulation introducing a third harmonic component into the output phase voltage [Boys and Handley, 1990]. The output voltage also has a low harmonic content, particularly at high modulation indices and where the ratio of the switching frequency to the fundamental output frequency is low [Holtz, 1994].

Techniques like sub-harmonic PWM and switching-frequency optimal PWM can be implemented using analogue electronics, however these analogue implementations have problems with balancing the three phases and with DC offsets. Digital waveform generation techniques solve these problems. SVM is a high performance digital technique, but it becomes relatively complex as the number of inverter levels increases.

4.2 Three-Level SVM Implementation

There are two stages to a SVM implementation, the selection of the voltage vectors and the calculation of their dwell times. There are various methods that can be used to achieve this. The technique used for this project is described in this section, while alternative SVM implementations are described in Section 4.3. Experimental results that show the performance of space vector modulation for a three-level inverter are provided in chapter 8.

4.2.1 Voltage Vector Selection

Minimal current distortion results if the nearest three voltage vectors to the reference vector are used. Figure 4.1 shows a plot of all the voltage vectors that the three-level inverter can produce, with an example of the reference vector, V^*, superimposed.

Figure 4.1 shows the vector diagram split into six sectors, identical to those for two-level space vector modulation. Within each of these six sectors are four triangles formed by the voltage vectors that lie at each vertex. These triangles will be termed sub-sectors. In Figure 4.1 the reference vector lies in Sector 1, Sub-Sector 2.

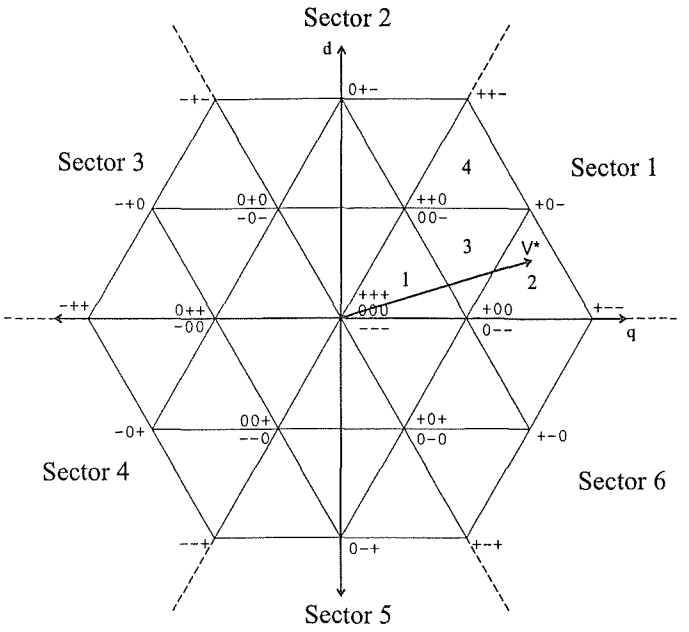


Figure 4.1: Vector Diagram

To determine the sector location, three geometric comparisons are performed. The first two comparisons determine the quadrant of the d,q vector space, while the third comparison differentiates between the two sectors that lie within each quadrant. Once the sector is determined the reference vector is rotated back into sector number 1, similar to two-level space vector modulation. This simplifies the determination of the sub-sector location.

Three more geometric comparisons are performed to determine the sub-sector location within that sector. Each comparison divides the sector into smaller regions while searching for the region that contains the reference vector. The sector and sub-sector numbers are then used as indices to a lookup table that contains the voltage vectors that reside at the vertices of each sub-sector.

4.2.2 Switching Sequence

The sequence of the three nearest vectors is chosen such that only one phase leg of the inverter is required to switch in order to change from one vector to the next. This ensures that the minimum possible device switching frequency is used.

The switching sequence always starts with one of the voltage vectors that can be generated using multiple inverter states. One phase will switch to move to one of the two adjacent vectors. Another phase will then switch to move to the next adjacent vector. When the last phase switches the inverter has returned to generating the first vector but with all the phases in a different state. The process then repeats in reverse to switch back to the original inverter state.

An example is shown in Figure 4.2 for a reference vector that lies in sub-sector 2 of sector 1. The sequence starts at 0-- (V_1) and phase A switches from the neutral point to the positive rail to get to +-- (V_2). Phase B switches from the negative rail to the neutral point to generate +0- (V_3). Phase C switches from the negative rail to the neutral point to generate +00 (V_1). The same voltage vector is being applied to the motor except the inverter is now in a different state. All of the three nearest vectors to the reference vector have been applied once. This sequence of vectors is called a sub-cycle. The next sub-cycle is in reverse, phase C switches back to the negative rail to get +0-, phase B switches back to the negative rail to get +--, phase A switches back to the neutral point to get 0--, and the inverter is back in its original state.

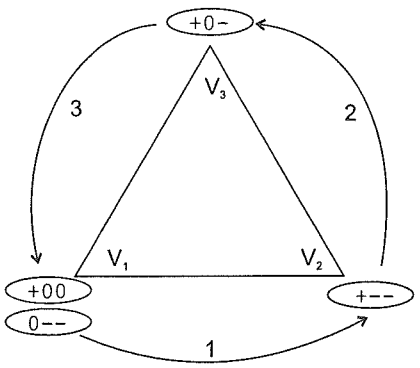


Figure 4.2: Switching Sequence

After two sub-cycles each phase has switched from its original state to a different state and back again. This is demonstrated by Figure 4.3, which shows the sequence of vectors used in a single switching cycle. Each switching cycle is composed of two sub-cycles, one the reverse sequence of the other. V_1 is divided evenly between the start and the end of the sub-cycle and is always the zero vector or one of the six inner voltage vectors that has two possible inverter states. By correctly choosing the switching sequence, it is

possible to apply each voltage vector twice and have each phase leg perform one switching cycle.

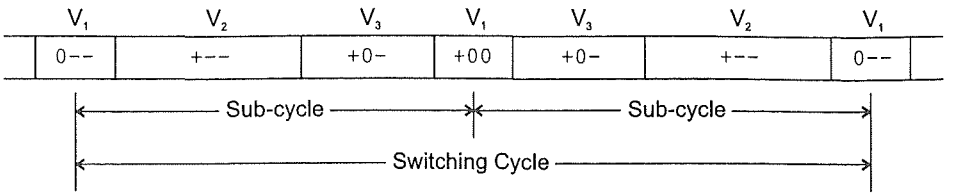


Figure 4.3: Switching Cycle

4.2.3 Neutral Point Voltage Control

The preceding section showed that all of the switching sequences start with the voltage vector that can be produced using multiple inverter states. Section 2.1.4 showed that these duplicate inverter states affect the neutral point of the DC bus in different ways. It is important that the neutral point voltage is controlled such that it remains at the mid-point between the positive and negative DC bus rails, since a voltage unbalance results in the generation of second harmonic components in the output voltage [Liu, Choi et al., 1991].

In the preceding section the first voltage vector, V_1, was divided evenly between the two different inverter states. Instead of dividing the time evenly, the neutral point voltage can be controlled by adjusting the ratio between the two inverter states, as long as the total dwell time of the voltage vector remains fixed. For example, if the neutral point voltage starts to decrease, the inverter state that would cause it to increase can be used to keep the DC bus balanced.

Instead of controlling the neutral point voltage by varying the ratio of the two inverter states, it can be controlled by the appropriate selection of the inverter states over a number of switching cycles. This method is used to control the neutral point voltage in this project as it is slightly simpler. Sub-sectors 2 and 4 in Figure 4.1 each only contain one voltage vector that can be achieved using multiple inverter states and that vector is always selected such that the two states are split evenly between the start and end of the sub-cycle. There is no control over the neutral point voltage while the reference vector

lies in these regions. Sub-sectors numbered 1 and 3 have more redundancy and allow the neutral point voltage to be controlled.

Figure 4.4 shows the two possible switching sequences for sub-sectors each of 1 and 3. For sub-sector 1, diagram a) shows a sequence of inverter states that only use the lower half of the DC bus and therefore cause the neutral point voltage to decrease. Diagram b) shows a sequence of inverter states that only use the upper half of the DC bus and therefore causes the neutral point voltage to increase.

Diagrams c) and d) in Figure 4.4 show the two possible sequences for sub-sector 3. Two of the three voltage vectors used in sub-sector 3 can be produced using multiple inverter states. One of these vectors will be split evenly between the start and end of the sub-cycle while the other vector will be used for varying lengths of time depending on the exact location of the reference vector. In diagram c) the sequence is chosen such that +00 and 0-- are used evenly, while 00- will decrease the neutral point voltage. Diagram d) is the opposite; ++0 and 00- are used evenly, while +00 will increase the neutral point voltage.

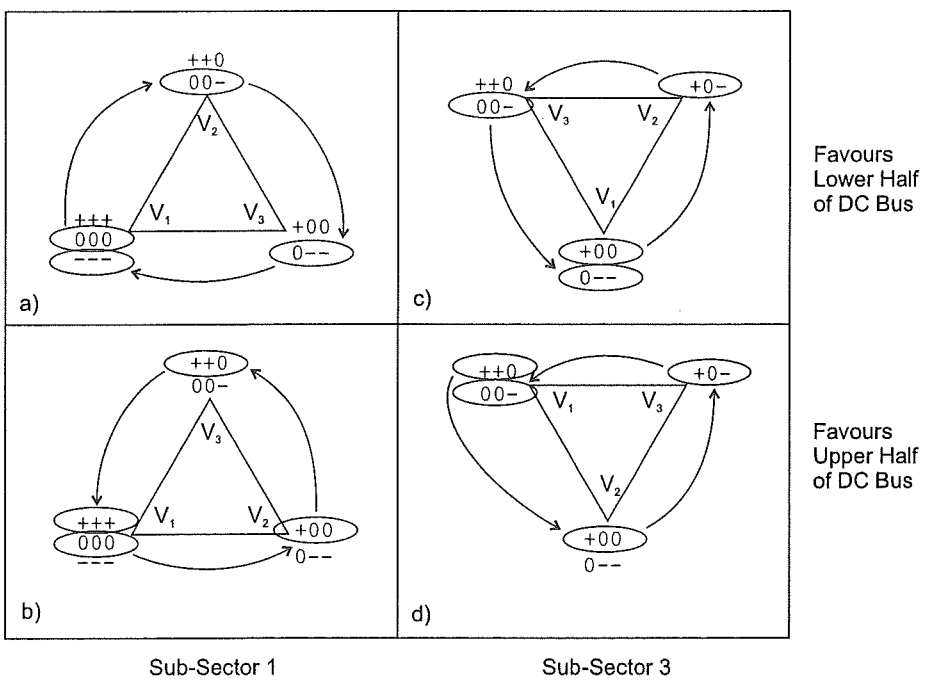


Figure 4.4: Switching sequences for neutral point voltage control

Sub-sectors 2 and 4 have no control over the neutral point voltage since the two inverter states that affect the neutral point are used evenly. There are two possible switching sequences for sub-sectors 1 and 3. One sequence will increase the neutral point voltage while the other will decrease it. Normally the two sequences are used on alternate switching cycles to approximately cancel their effects. If an unbalance between the two halves of the DC bus is detected, one of the two sequences can be used continuous until the unbalance is corrected. The lookup table mentioned in Section 4.2.1 that determines the nearest three vectors also contains information on the possible switching sequences and how they affect the neutral point. The lookup table is explained in more detail in Section 7.3.5 where the software implementation is described.

4.2.4 Dwell Time Calculation

The proportion of the total switching cycle that a vector is used for is called its duty-cycle. By applying each of the three nearest vectors for an appropriate duty cycle, they can be averaged over a fixed switching cycle to obtain the reference vector. This process can be expressed in matrix form as equation (4-1) which shows the reference vector, $\overline{V^*}$, is equal to the three nearest vectors ($\overline{V_1}$, $\overline{V_2}$, $\overline{V_3}$) multiplied by their respective duty cycles (d_1, d_2, d_3).

$$\begin{bmatrix} V_q^* \\ V_d^* \end{bmatrix} = \begin{bmatrix} V_{1q} & V_{2q} & V_{3q} \\ V_{1d} & V_{2d} & V_{3d} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{4-1}$$

There is the second requirement that the three vectors must fill the complete cycle. This requirement can be expressed as equation (4-2).

$$d_1 + d_2 + d_3 = 1 \tag{4-2}$$

Equations (4-1) and (4-2) can be combined to create equation (4-3), a single expression for the reference vector in terms of the three nearest vectors and their duty cycles. Equation (4-3) can be converted into (4-4), an expression for the duty cycles in terms of the chosen three vectors and the reference vector.

$$\begin{bmatrix} V_q^* \\ V_d^* \\ 1 \end{bmatrix} = \begin{bmatrix} V_{1q} & V_{2q} & V_{3q} \\ V_{1d} & V_{2d} & V_{3d} \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{4-3}$$

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} V_{1q} & V_{2q} & V_{3q} \\ V_{1d} & V_{2d} & V_{3d} \\ 1 & 1 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} V_q^* \\ V_d^* \\ 1 \end{bmatrix} \tag{4-4}$$

In practice, the voltage vectors $\overline{V_1}$, $\overline{V_2}$, and $\overline{V_3}$ are determined each switching cycle using the chosen switching sequence and the instantaneous DC bus voltage. Equation (4-3) is then solved in real-time as described in Section 7.3.5.

4.2.5 Summary of SVM Implementation

Sections 4.2.1 to 4.2.4 have described the steps required to implement a space vector modulator. The reference vector is synthesised using the three nearest vectors to the reference vector that the inverter can produce. When multiple inverter states exist for one or more of the voltage vectors, the appropriate states are selected to control the voltage of the DC bus neutral point. The switching sequence of the states is selected such that only one phase leg is required to switch in order to transition from one vector to the next. The dwell time of each of the three vectors are calculated in order to approximate the reference vector over a fixed switching period.

4.3 Alternative Implementations

Various techniques have been published that describe different ways to perform the required calculations or handle the neutral point voltage-balancing problem [Liu, Choi et al., 1991], [Zhang, 1995]. Also, since high voltage three-level inverters are commonly implemented using GTOs, some work has been done to modify SVM to satisfy the minimum GTO on-time requirements [Liu and Cho, 1994], [Lee, Suh et al., 1996].

Since the space vector modulator described in Section 4.2 was implemented, a new technique has been published that could enable multi-level SVM to be performed with reduced microprocessor execution time [Seo, Choi et al., 2001]. This new technique

converts the three-level SVM problem into an equivalent two-level SVM system by considering the three-level d,q vector space as being composed of six smaller hexagons, each centred on one of the six inner voltage vectors. The switching sequence and dwell times are calculated for that smaller hexagon and the results are converted back to a three-level system. The processor time required to perform the transformations is offset by the reduced calculation time for the simpler two-level system. The reduced processor requirements may be an important consideration in a commercial implementation.

4.4 Effect of Inverter Dead-Time

In Section 4.2 the effect of device switching delays were ignored. In practice, IGBTs take a few hundred nanoseconds to a few microseconds to switch on or off. Because of these restrictions, a short delay called dead-time is inserted between one set of IGBTs turning off and the next set turning on. This delay ensures both sets of IGBTs are not on at the same time, causing a short circuit across the DC bus.

In addition to the dead-time, a minimum on-time is also enforced. The dead-time is inserted because it takes an IGBT a fixed length of time to turn-off. The minimum on-time requirement exists because it takes an IGBT a fixed length of time to turn-on. If the turn-on time of the IGBT is greater than the required SVM on-time, the IGBT will not have fully turned on before the gate signal is removed.

The following sections analyse the effect that the minimum on-time and dead-time requirements of the inverter have on the output voltage and how that effect can be compensated.

4.4.1 Minimum Device On-Time Limitation

If the calculated dwell time of any vector is less than a configured minimum, that vector is discarded. The actual time that the IGBT gate signal is active for is the calculated dwell time minus the dead-time. Therefore, if a minimum gate signal of 1μs is desired, and the dead-time is also 1μs, then any vectors with a calculated dwell time less than 2μs can not be used.

The short amount of time that the vector would have been applied for is redistributed evenly between the two remaining vectors in order to maintain a fixed switching period. If there are two vectors below the minimum on-time limit, both are discarded and the remaining vector is held for the complete switching cycle.

4.4.2 Dead-time Effect

During the dead-band all IGBTs in a phase leg are turned off. During this time current will continue to flow through the freewheeling diodes of the inverter. Figure 4.5 shows one phase leg of the inverter switching from the zero output state to the positive output state with an intermediate dead-band state. The upper set of diagrams show the current flowing from the load into the inverter (negative current with respect to the inverter) and the lower set shows the current flowing from the inverter into the load.

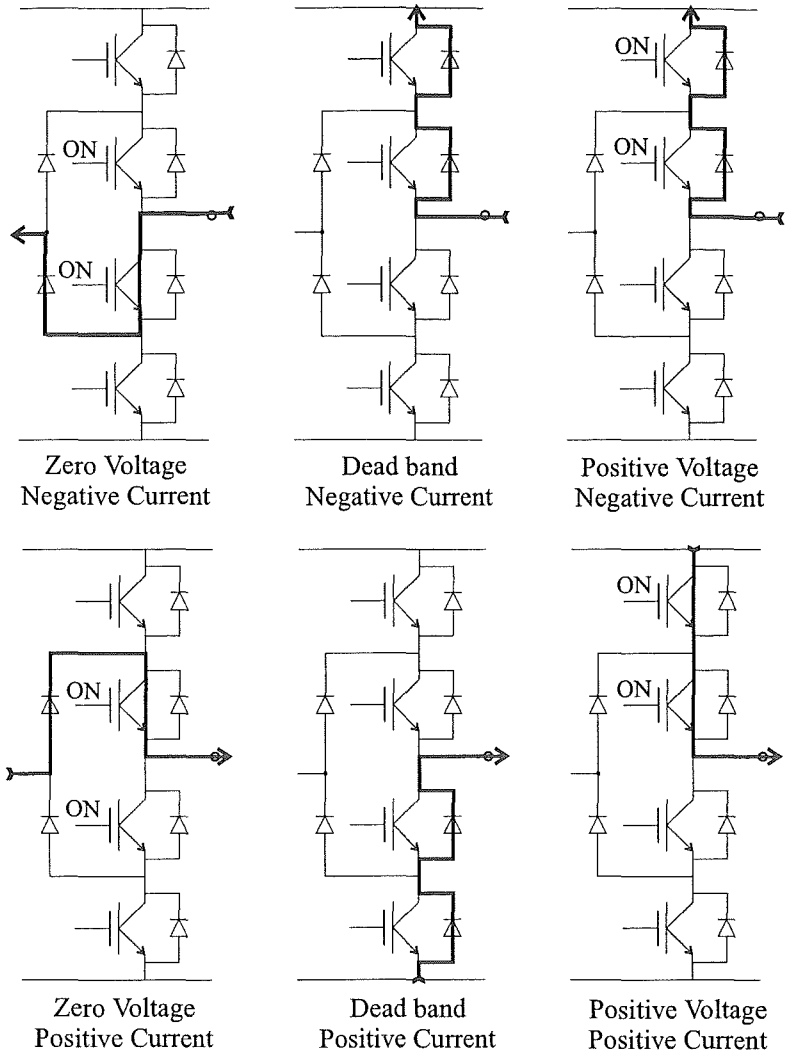


Figure 4.5: Current Flow during the dead-time

In the first of the upper three diagrams of Figure 4.5, the current is flowing through one of the inner IGBTs and its associated clamping diode to the neutral point. During the dead-band, the two inner IGBTs turn off and the current is conducted through the two upper freewheeling diodes. At the end of the dead-band the two upper IGBTs turn on, but the current keeps flowing through the freewheeling diodes. This shows that positive phase voltage is applied to the load during the dead-band.

In the first of the lower three diagrams of Figure 4.5, the current is again flowing through one of the inner IGBTs and associated clamping diode from the neutral point to the load. During the dead-band, the only path for the current is through the freewheeling diodes of the two lower IGBTs. At the end of the dead-band, the two upper IGBTs turn on and

conduct the current. This shows that a negative phase voltage is applied to the load during the dead-band.

Figure 4.5 shows if the load current is positive (from inverter to load), the phase voltage during the dead-band is negative. If the load current is negative, then the phase voltage is positive. The same results are obtained if all other switching combinations are examined.

In the case of a two-level inverter, the voltage during the dead-band must always be either the voltage of the current inverter state or the voltage of the next state. Therefore the effect of the dead-band is only to lengthen the on-time of one of the existing states [Jeong, Lee et al., 1988]. Figure 4.5 shows that a three-level inverter can generate a negative output voltage while switching from a zero output state to the positive output state. Figure 4.6 shows an oscilloscope plot showing the phase voltage going negative during the 1μs dead-band when the output switches from zero to positive or positive to zero.

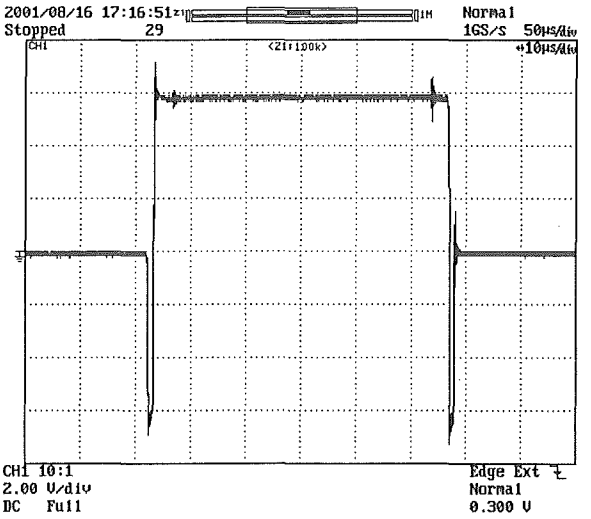


Figure 4.6: Output phase voltage

The phase voltage during the dead-band is only dependant on the load current direction and the PWM switching pattern. The magnitude of the effect is only dependant on the length of the dead-time relative to the length of the switching cycle. As the magnitude of the generated output voltage decreases, the relative effect of the dead-time becomes larger since the dead-time becomes a significant fraction of the vector dwell time.

The dead-time effect results in a distorted fundamental output voltage that becomes even more distorted at low amplitudes. The output voltage contains low order harmonic components that cause increased copper losses, iron losses, and torque pulsations [Murai, Watanabe et al., 1987].

Some induction motor control schemes such as sensor-less vector control and direct torque control require knowledge of the applied stator voltage in order to estimate motor speed or stator flux. The output voltage is normally determined from the applied switching sequence and the DC bus voltage. The error in the output voltage introduced by the dead-bands causes errors in the speed estimate or output torque [Liu and Chen, 1998].

The Direct Torque Control technique used for this project estimates the induction motor stator flux from the current, stator resistance, and applied voltage. Errors in the applied voltage contribute to problems with the already difficult stator flux estimation. Therefore, some form of dead-time compensation is required to enable accurate torque control. Various techniques have be developed for two-level inverters that either correct the IGBT gate timing in hardware [Jeong, Lee et al., 1988] or correct the PWM timing calculations [Oh, Kim et al., 1995]. None of these techniques can be directly applied to a three-level inverter due to the larger number of possible switching sequences and therefore increased complexity.

Various techniques for the dead-time compensation in a three-level inverter are possible.

1. Correct the SVM reference vector based on an estimate of the dead-time effect
2. Correct the SVM reference vector based on the actual effect of the dead-time on the preceding switching cycle.
3. Feed back an estimate of the actual output voltage to the stator flux estimator so that the torque controller can compensate for the dead-time.

The third technique was chosen based on its robustness and ease of implementation.

4.4.3 Dead-time Compensation for a Three-Level Inverter

The actual output voltage is calculated from the switching pattern used and the measured DC bus voltage, taking into account the dead-time and minimum on-time effects. The

output voltage ignoring the effect of the dead-bands is first calculated and then the effect of the dead-band for each commutation is included. The calculation of the output voltage is achieved using the following steps.

1. For each of the three voltage vectors that were used, the active duty cycle, d_x, is calculated using (4-5) such that the dead-time, t_d, is ignored. In one switching cycle of t_s, each vector is applied twice for the duration t_x. The duty cycle will be zero if the vector was discarded due to being less than the minimum on-time. The output voltage ignoring the dead-time is then calculated using (4-6), where $\overline{V_1}$, $\overline{V_2}$, and $\overline{V_3}$ are the voltage vectors produced by each of the inverter states that were used. These are calculated using a lookup table for each inverter state and the measured DC bus voltage.

$$d_x = \frac{2 \cdot (t_x - t_d)}{t_s}, \text{ where } x = 1,2, \text{ or } 3 \tag{4-5}$$

$$\begin{bmatrix} V_{outq} \\ V_{outd} \end{bmatrix} = \begin{bmatrix} V_{1q} & V_{2q} & V_{3q} \\ V_{1d} & V_{2d} & V_{3d} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{4-6}$$

2. The output state of each phase during the dead-band is determined from the measured load current. If the current in any phase is positive, the voltage will be negative and visa versa.
3. The magnitude of the dead-band voltage for each commutation is calculated using (4-7). Each phase will change states twice in each switching cycle, requiring two dead-band periods. As the dead-time increases or the length of the switching cycle decreases, the effect of the dead-band becomes worse.

$$|V| = \frac{2t_d}{t_s} \cdot \frac{2}{3} V_{dc} \tag{4-7}$$

4. For each voltage vector transition that occurred, the following steps are performed to calculate the effect that the dead-band had on the output voltage;
   4.1. The phase or phases that switched during that voltage vector transition are determined. If an intermediate vector was dropped due to the minimum on-time limitation, it is possible that two phases will switch simultaneously.
   4.2. The output state of the phase(s) that switched are replaced with the dead-band output state calculated at step 2.

4.3. The calculated output states of each of the three-phases are converted into a vector of unit length using a lookup table.

4.4. The vector is scaled to the correct magnitude using the voltage computed in step 3. This gives the effective voltage vector that was applied during the dead-band.

4.5. The calculated dead-band voltage is added to the output voltage from step 1. This process is repeated for each vector transition that occurred.

5. The final voltage estimate is then fed back to the stator flux estimator of the torque controller, which then compensates for the dead-time. The torque controller ensures that the stator flux vector follows the required trajectory by producing whatever voltage is necessary.

The calculated output voltage is only an estimate. It was assumed the length of time that the inverter dead-band states were active for was equal to the length of the dead-time between the applied gate signals. In practise this is not completely valid due to the unsymmetrical switch-on / switch-off characteristic of the IGBTs. This is difficult to compensate for as the load current and temperature affect the IGBT switching characteristics.

4.4.4 Measured Results

The output voltage estimation algorithm was tested with a real inverter and an induction motor as a load using the experimental set-up described in chapter 8. This consisted of a three-level IGBT inverter switching at 10kHz with a 1μs minimum on-time and 1μs dead-time. A 240V DC bus was used with an induction motor as the inverter load. Space vector modulation was used to generate a sinusoidal output voltage. In a d,q vector space this corresponds a rotating vector with constant magnitude. Such a vector will have a circular trajectory.

The trajectory of the estimated output voltage vector was compared to the measured inverter output voltage vector. This measurement was obtained by filtering the three-phase inverter outputs and transforming them into a d,q reference frame using an op-amp based circuit. Sufficient filtering was included to remove most of the inverter switching whilst having minimal effect on the voltage vector trajectory.

Figure 4.7 shows d,q vector plots of the estimated and measured output voltage trajectory. Plots are shown for modulation indices of 0.06, 0.1, 0.2, and 0.8. A modulation index of one is defined as the maximum output voltage that can be generated using space vector modulation without employing over-modulation techniques. The estimated voltage plots for modulation indices of 0.06 and 0.1 also show the reference voltage vector. Due to the phase shift introduced by the circuit measuring the actual output voltage, the two sets of plots cannot be compared directly.

Figure 4.7 shows a close comparison between the estimated and measured output voltage trajectories, especially for modulation indices of 0.1 and above. These plots show that dead-time has a significant effect for voltages below 10% of the maximum voltage and has very little effect on high output voltages. At low voltages, the dead-time effect can be considered as a phase dependant voltage drop.

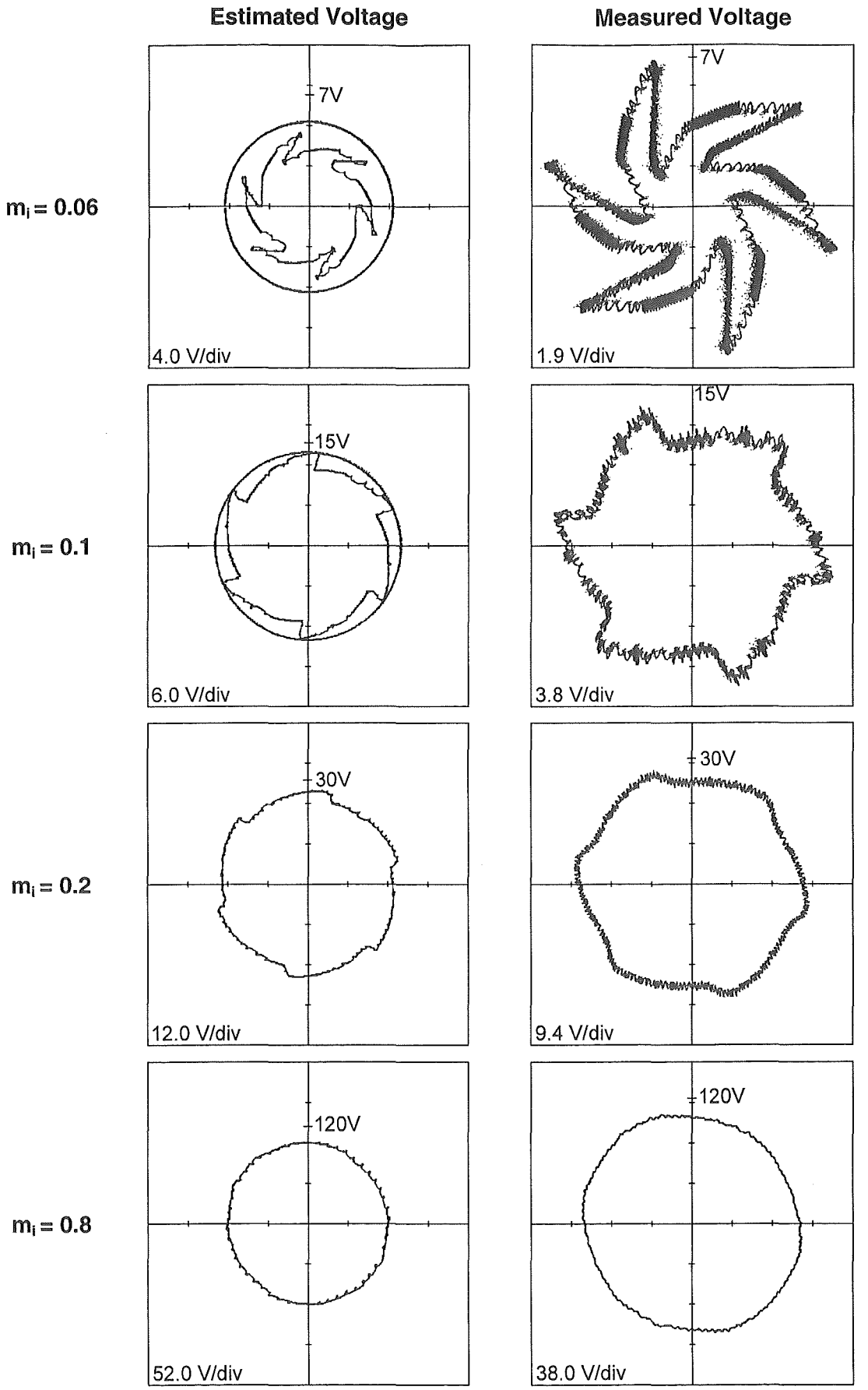


Figure 4.7: Effect of dead-time on the output voltage

4.5 Summary

This chapter has provided an overview of space vector modulation, a detailed description of the implementation used for this project, and the effect of inverter dead-time on the inverter output voltage. A summary of the SVM implementation used was provided in Section 4.2.5 and an overview of alternative implementations in Section 4.3.

This chapter showed that the effect of the inverter dead-time can be considered as a phase dependant voltage drop. The magnitude of the voltage drop is only dependant on the length of the dead-band relative to the length of the switching cycle. The relative effect of the dead-band becomes more significant as the generated output voltage decreases. The error in the inverter output voltage affects the stator flux estimated by the voltage model stator flux observer. By feeding an estimate of the actual output voltage back to the stator flux observer, the torque controller can compensate for the effect of the dead-band.

5. Simulation Results

The previous three chapters have described four different torque control schemes. These were field oriented control, direct torque control, minimal torque ripple DTC, and DTC using space vector modulation. The implementation of DTC using space vector modulation was described in detail, as it is the torque control scheme used for this project. This was chosen based on the results of computer simulations. These simulation results are presented in this chapter, along with a comparison of the advantages and disadvantages of the four control schemes.

Section 5.1 describes the models of the control schemes that are created to enable computer simulation. Section 5.2 presents torque, stator flux, and current waveforms that show the performance of the four control schemes under steady-state conditions. Section 5.3 presents the torque and flux waveforms during step changes in the torque and flux reference values. Finally, Section 5.4 summarises the simulation results and presents a comparison of the four control schemes to show why DTC using space vector modulation is chosen.

5.1 Simulation Models

Each of the four torque control schemes are simulated using MATLAB / Simulink. The parameters of these simulations are set to match the experimental system described in chapter 8. The top-level block diagrams of the four simulation models are included in Appendix A. Also included is the model of the field oriented controller and the top-level diagram of the induction motor model.

The model of the induction motor used for simulation is based on the induction motor model described in Section 2.2. The inputs to the simulation model are the applied voltage and rotor speed. The model then calculates the stator current, stator flux, rotor current, rotor flux, and torque. The simulation model is carefully arranged to avoid algebraic loops. These are created when the output of a block is directly dependant on an input and that input is obtained via a feedback path directly from the output of that block. These algebraic loops can be avoided by arranging the model such that there is an integral

in each of these feedback paths. The parameters of the motor model match the 400Hz motor that is used for the experimental testing described in chapter 8. These parameters are listed in Appendix E.

The four torque control models assume an ideal three-level inverter switching at 10kHz and supplied from a perfect 300V DC voltage source. The inverter is simulated as being constructed from ideal semiconductor devices, which switch instantly and have no voltage drop. The dead-time and minimum-on time requirements are also ignored such that the average output voltage produced matches the reference voltage.

For the simulation results presented in this chapter, the induction motor speed is fixed at 2000RPM to enable a comparison with the experimental results. The motor parameters used in the torque controllers are identical to those of the simulated motor, i.e. there is no parameter detuning. For the direct torque control schemes, the stator flux and torque are taken directly from the output of the simulated motor. This means that no stator flux or torque estimator is required.

For the direct torque controller model, the widths of the torque and flux hysteresis bands are set at 0.5Nm and 0.005wb respectively. This gives an average switching frequency of around 10kHz under the simulation conditions of Section 5.2 and enables a comparison to be performed with the other control schemes that have a fixed 10kHz switching frequency. A 25μs zero-order hold is imposed on the voltage vector selected by the DTC look-up table in order to simulate calculation delays. In a practical direct torque control implementation, it is necessary to sample the currents and estimate the stator flux and torque before making a comparison with the hysteresis bands to determine if a new voltage vector is required. The simulation model therefore holds each voltage vector for a minimum of 25μs before a new one is selected in order to simulate these calculation delays.

5.2 Steady State Performance

The steady-state performance of the four torque control schemes is evaluated based on the torque and stator flux ripple, and on the current distortion. Sections 5.2.1 to 5.2.4 present

the torque, stator flux, and phase current waveforms and the phase current frequency spectrum.

All of the simulations are run until steady state operation is achieved with a torque reference of 5Nm, flux reference of 0.047wb, and at a fixed motor speed of 2000RPM. The three direct torque controllers use a stator flux reference of 0.047wb, whereas the field-oriented controller uses a rotor flux reference of 0.047wb, which results in a stator flux slightly larger than 0.047wb. About 6ms of the waveforms are shown in each section. This corresponds to a single cycle of the current.

5.2.1 Torque Ripple

Figure 5.1 shows the torque ripple for each of the four torque control schemes. Field-oriented control and DTC using space vector modulation both exhibit very low torque ripple of 0.3Nm. This is due to both schemes using space vector modulation to synthesise the voltage.

The average torque ripple for classical direct torque control was 0.8Nm. This is higher than the torque hysteresis bandwidth of 0.5Nm due to the 25μs voltage vector hold time that is used to simulate calculation delays. It is possible for the torque to leave the hysteresis band for a period of time before a new voltage vector is selected to drive it in the opposite direction, back inside the hysteresis band.

The average torque ripple for minimal torque ripple DTC is 1.2Nm, which is slightly larger than for classical DTC. This is due to the way minimal torque ripple DTC was extended to three-level operation. Under the operating conditions used for this simulation, classical DTC is only using the 6 inner voltage vectors whereas minimal torque ripple DTC always uses the 12 outer voltage vectors. Using the larger magnitude voltage vectors causes a greater change in the torque.

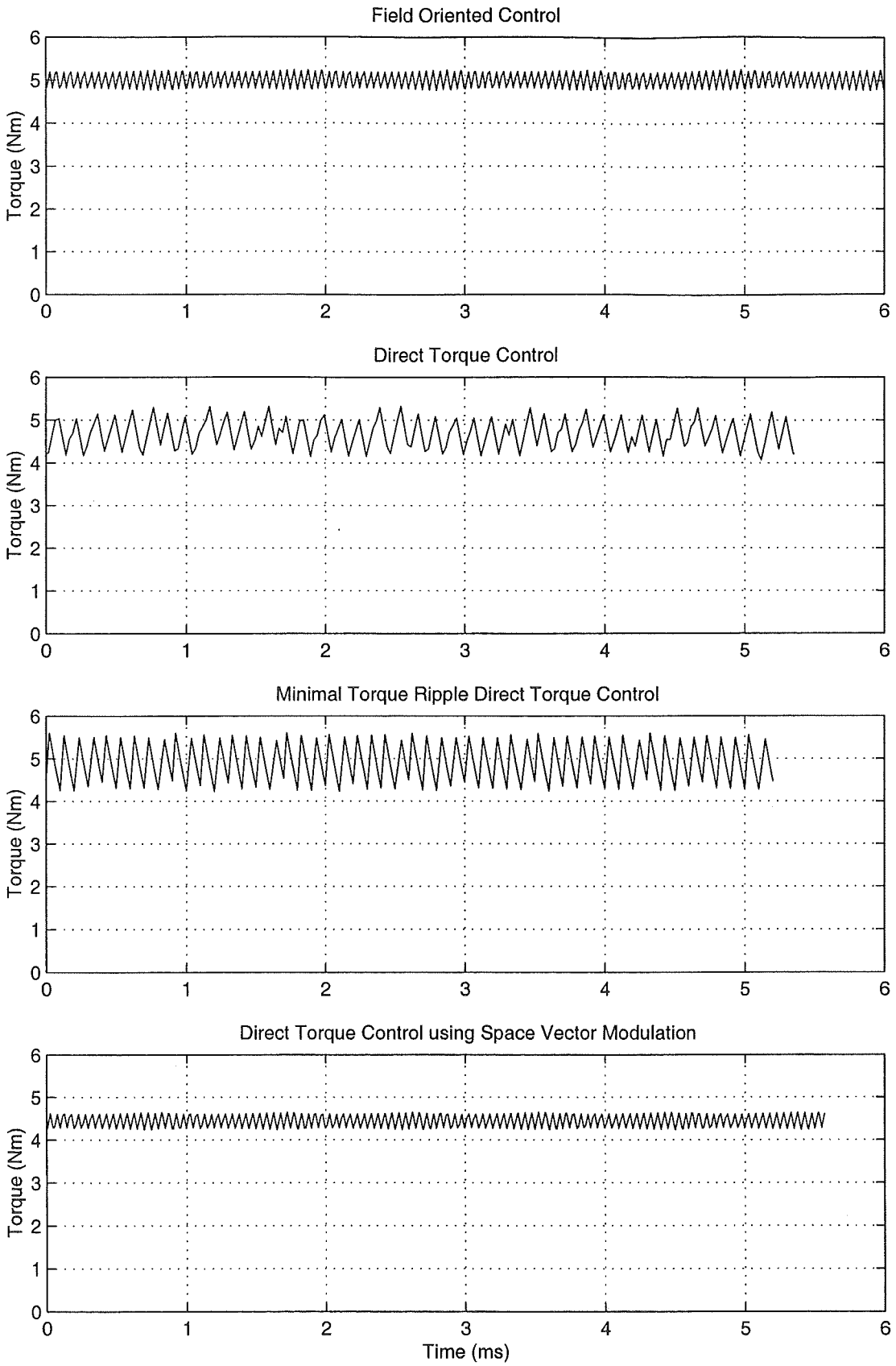


Figure 5.1: Comparison of the simulated steady-state torque waveforms

5.2.2 Stator Flux Ripple

Figure 5.2 shows a comparison of the stator flux ripple. Again, field-oriented control and DTC using space vector modulation produce very low ripple due to the space vector modulation technique used to synthesise the voltage. The stator flux ripple for these control schemes is 0.0013wb, which corresponds to a 2.7% variation in the reference flux value. The stator flux ripple for classical DTC is slightly larger at 0.002wb (4.3% of the flux reference) but is still at an acceptable level. The stator flux ripple for minimal torque ripple DTC is relatively high at 0.005wb (10.6% of the flux reference).

Minimal torque ripple DTC only considers torque ripple minimisation. In classical DTC, a new voltage vector is selected as soon as it is determined that the flux has left the hysteresis band. In minimal torque ripple DTC, a fixed switching cycle is used and the applied voltage vector duration is determined to minimise torque ripple without regard for the error in the flux.

The effect that the applied voltage has on the torque and flux depends on the phase difference between the stator flux vector and the applied voltage vector. When the stator flux is at the start of a sector, the phase difference is larger than when the stator flux is at the end of the stator. A larger phase difference means a greater effect on torque and a smaller effect on the stator flux magnitude. If the effect of the applied voltage on the torque is smaller, the torque ripple minimisation stage calculates that it needs to be applied for a longer duration.

At the end of a sector, the phase difference between the stator flux and selected voltage vector is smaller. Therefore the effect on stator flux is larger, and because the effect on torque is smaller, the voltage is held for a longer duration to get the same increase. The combination of a larger effect on the flux magnitude and longer on-duration results in relatively large changes in stator flux magnitude for minimal torque ripple DTC when the stator flux lies near the end of a sector.

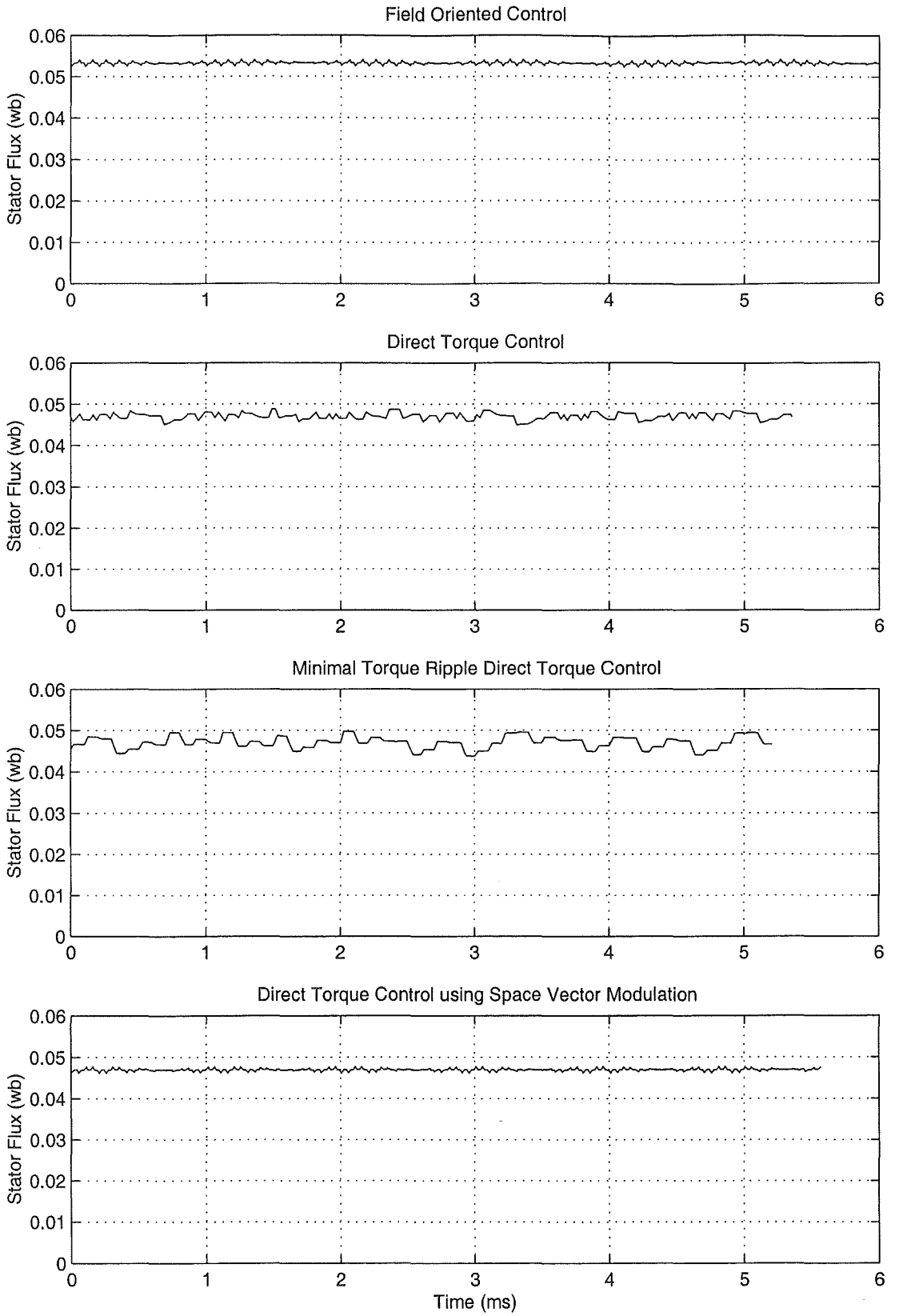


Figure 5.2: Comparison of the simulated steady-state stator flux waveforms

5.2.3 Current Ripple

Figure 5.3 shows a comparison of the simulated phase current waveforms over a single period of the current. The simulated current waveform and the fundamental component are shown for each plot. The current period for field oriented control is slightly longer as a lower excitation frequency is required due to the slightly higher flux levels. Again, field-oriented control and DTC using space vector modulation produce very low ripple due to the space vector modulation used to synthesise the voltage. Classical DTC is slightly more distorted and minimal torque ripple DTC has the most distorted current waveform.

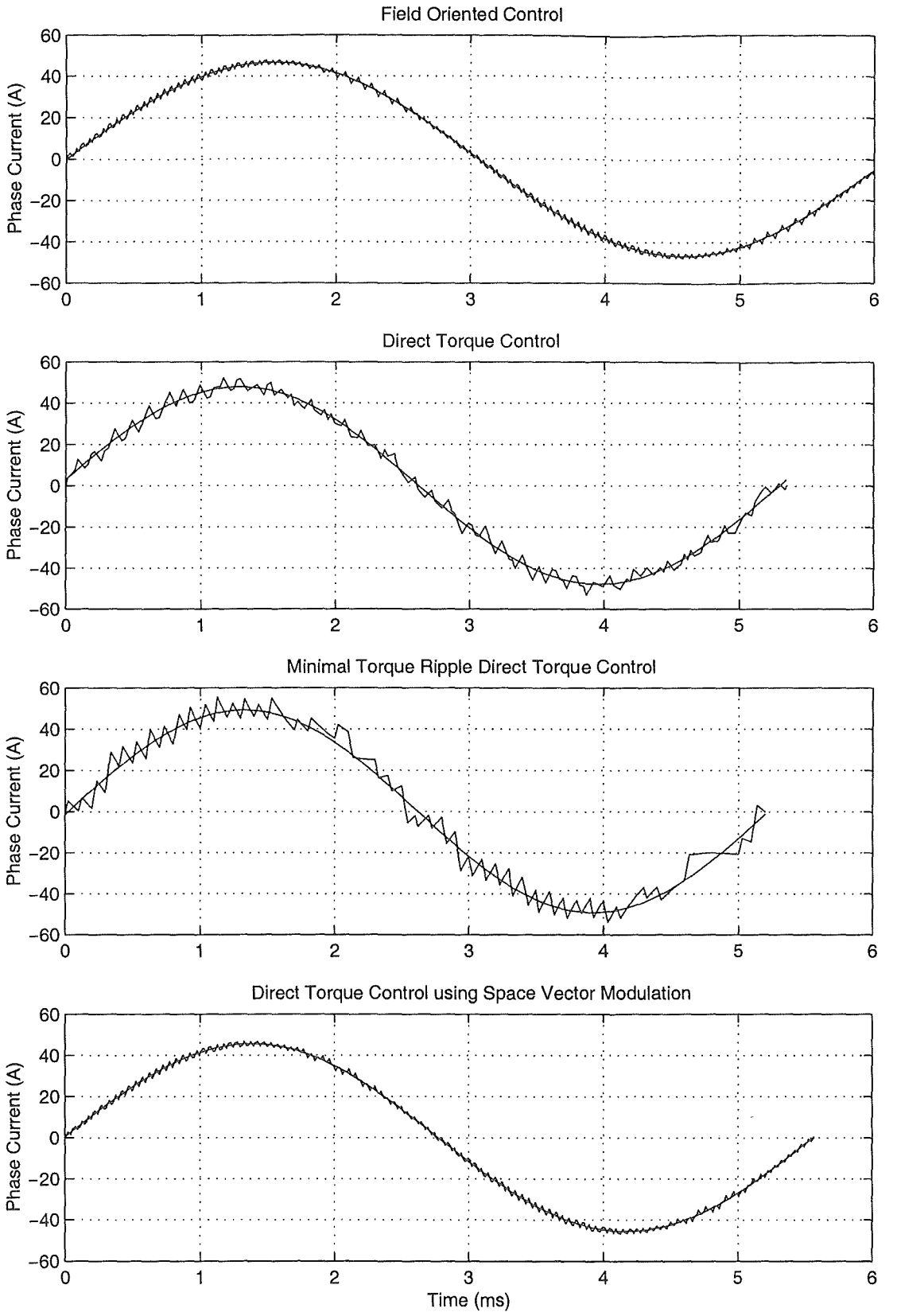


Figure 5.3: Comparison of the simulated steady-state phase current waveforms

5.2.4 Current Frequency Spectrum

Figure 5.4 shows a comparison of the frequency spectrums of the current waveforms of Figure 5.3. These are over the range of 0 to 20kHz such that the switching frequency is included. Due to the regular switching pattern of the space vector modulator, field oriented control and DTC using space vector modulation only show peaks near the switching frequency. For field-oriented control, the result of the modulation of the switching frequency with the fundamental, 5^{th}, 7^{th}, 11^{th} and 13^{th} harmonics can be clearly seen. DTC using space vector modulation also shows harmonics at 5kHz and 15kHz due to a more realistic model of the three-level space vector modulator being used that takes into account the equalisation of the DC bus. The cause of these additional switching harmonics is described in detail in Section 8.2.3.

Classical direct torque control does not have a fixed switching pattern and results in a wide spectrum of current harmonics. Minimal torque ripple DTC uses a fixed 100μs switching cycle but does not have a periodic switching pattern. The result is a wide spectrum of current harmonics plus two peaks centred on 10kHz due to the modulation of the fundamental component with the fixed switching cycle.

The resulting total harmonic distortion (THD) is shown in Figure 5.4 for each of the control schemes. The THD is taken over all harmonics of the fundamental component and is expressed as a percentage of the fundamental component. Field-oriented control and DTC using space vector modulation are around 2.6%, classical DTC is 6.7%, and minimal torque ripple DTC is 11.2%.

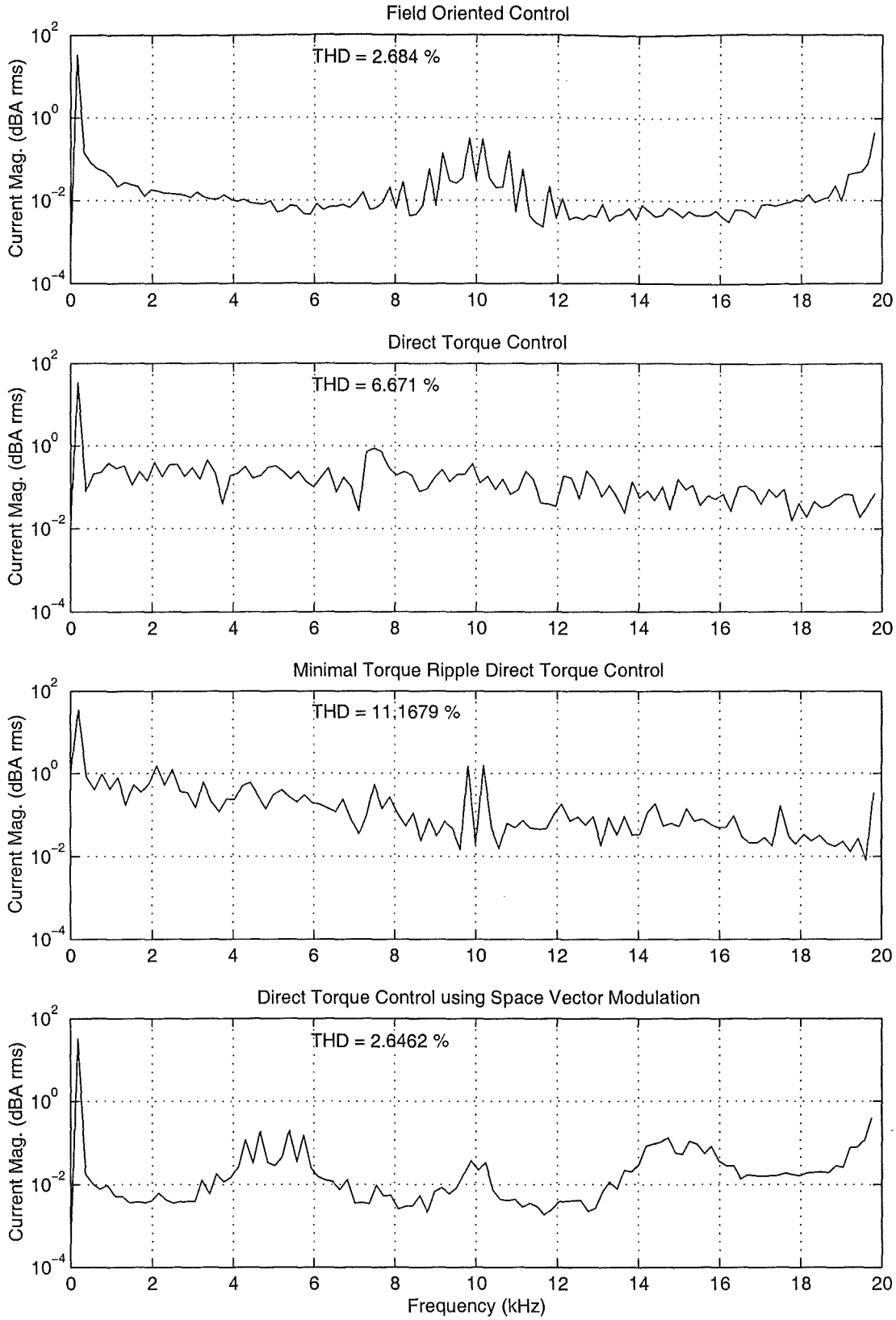


Figure 5.4: Comparison of simulated current frequency spectrums

5.3 Transient Performance

The transient responses of the four control schemes are evaluated by simulating step changes in the torque and flux references. Sections 5.3.1 to 5.3.4 present the results for each of the four torque control schemes. Each controller is simulated with a flux reference of 0.03wb and a torque reference of 1Nm until steady state operation is reached. The flux reference is then stepped from 0.03wb to 0.05wb after 4ms and the torque reference is stepped from 1Nm to 6Nm after 8ms. A DC bus voltage of 300V is used and the motor speed is fixed at 2000RPM.

5.3.1 Field Oriented Control

Figure 5.5 shows the response of the field-oriented controller to step changes in the torque and flux references. Because of the relatively slow response, a longer time scale is used compared to the other control schemes. At 10ms the flux is stepped from 0.03wb to 0.05wb and at 20ms the torque reference is stepped from 1Nm to 6Nm. Figure 5.5 shows the torque, rotor flux magnitude, and the q and d axis components of the stator current in the rotor reference frame.

When the flux reference is changed at 10ms, there is a step-change in the d-axis stator current reference and the PI controller regulates the voltage to drive the actual stator current to the new reference value. Due to the higher d-axis stator current, the rotor flux increases slowly with the rotor time constant and the q-axis current reference decreases slightly as less current is required to produce the same torque at the higher level of flux.

At 20ms there is a step change in the torque reference that causes a step change in the q-axis current reference, an increase in the actual q-axis current, and therefore an increase in torque. The rotor flux takes 6ms to increase to the new flux reference value and the torque takes 1.5ms to increase to the new torque reference value.

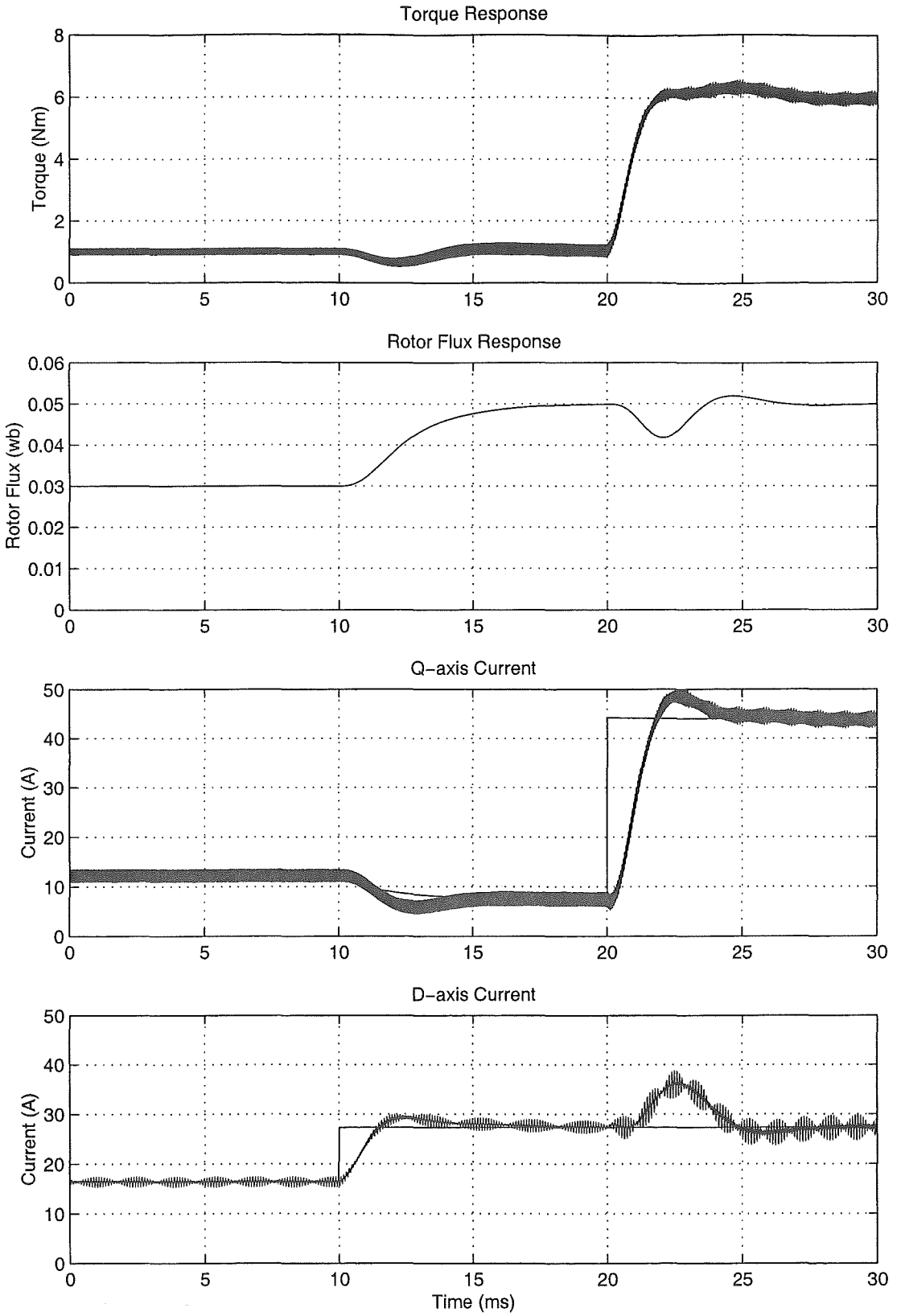


Figure 5.5: Transient performance of field-oriented control

5.3.2 Direct Torque Control

Figure 5.6 shows the transient response of the direct torque control scheme. At 4ms the stator flux reference is stepped from 0.03wb to 0.05wb. Figure 5.6 shows the actual stator flux has risen to the new reference value within 1.6ms. During this time the controller selects voltage vectors that increase the flux while maintaining control over torque. The controller selects a non-zero voltage vector to increase the torque and flux. When the torque has reached the upper edge of the hysteresis band, the zero voltage vector is selected to allow the torque to slowly decrease. While the zero voltage vector is applied, there is virtually no change in the stator flux magnitude. The result can be observed in the flux magnitude plot as a series of small steps as the controller alternates between the non-zero and zero voltage vectors.

The step change in the torque reference occurs at 8ms. Figure 5.6 shows that the torque produced by the motor increases to the new torque reference within 150μs. Figure 5.6 also shows there is complete decoupling between the torque and flux. There is no change in the torque during the step change in the flux reference, and no change in the flux during the step change in the torque reference.

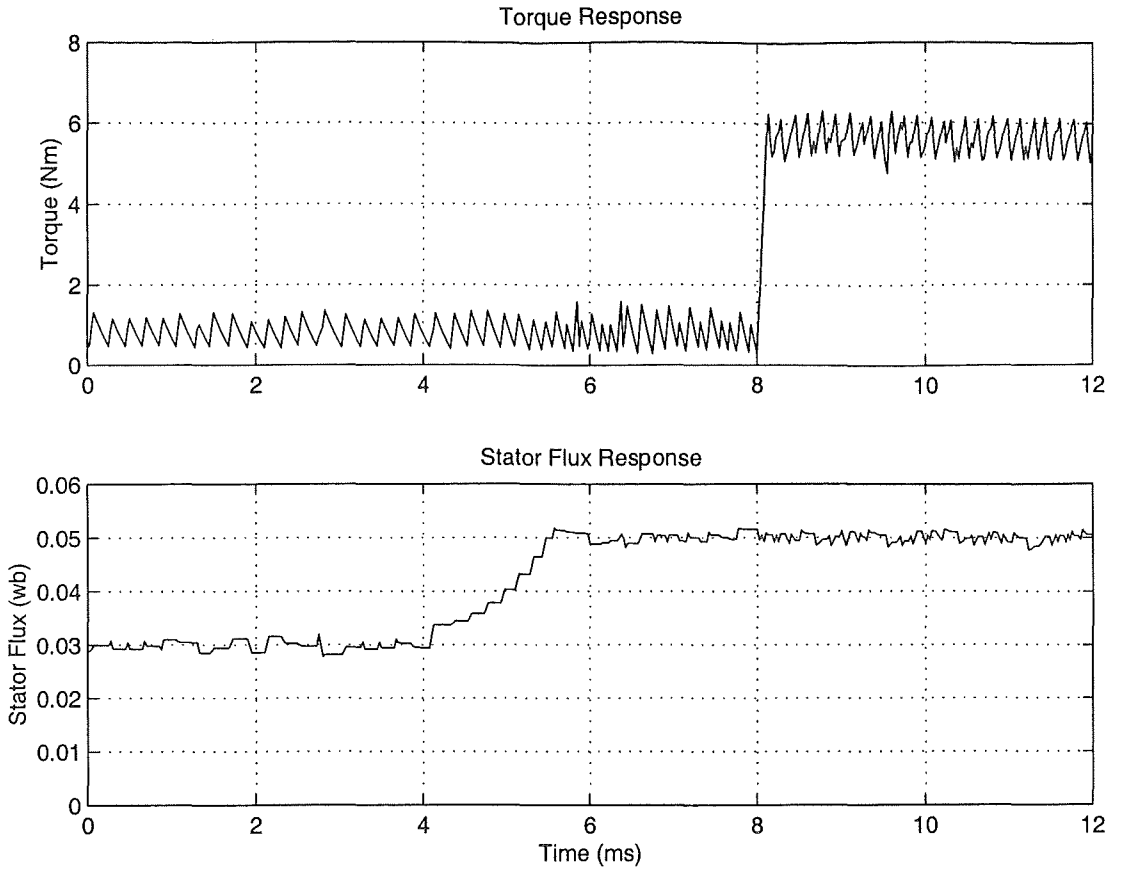


Figure 5.6: Transient performance of direct torque control

5.3.3 Minimal Torque Ripple DTC

Figure 5.7 shows the transient response of minimal torque ripple DTC. Like classical DTC, the step change in flux occurs as a series of small steps, but takes slightly longer at 2.7ms. The torque change takes a single 100μs switching period to increase to the new reference value. Compared to classical direct torque control, Figure 5.7 shows reduced torque ripple when operating at low levels of flux and torque but higher torque ripple when operating at higher levels of flux and torque.

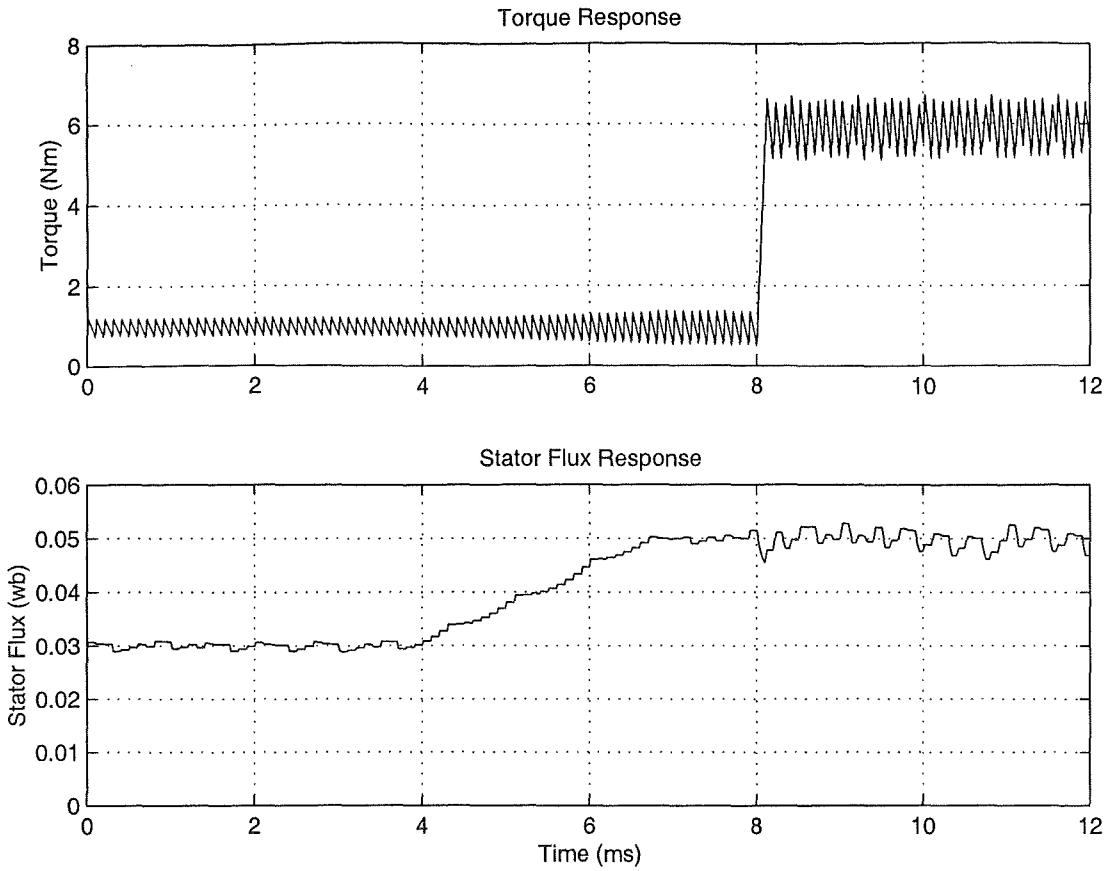


Figure 5.7: Transient performance of minimal torque ripple DTC

5.3.4 DTC using Space Vector Modulation

Figure 5.8 shows the transient performance of DTC using space vector modulation. The stator flux takes 200μs (two switching cycles) to increase to the new reference value. During this time there is a wide variation in the torque. At the time of the change in flux reference, the torque is below the reference value and therefore a voltage vector is selected to increase the torque and flux. During the first 100μs switching cycle, the torque increases by 3Nm. On the next cycle a voltage vector is selected to increase flux while decreasing torque and the torque drops by almost 3Nm. When the flux has reached the reference value, full control of torque and flux is restored. This short burst of torque is not an issue in a practical application since a step change in the flux reference is not normally required. If a step change were required, the burst of torque would not be noticeable since it only lasts for 200μs.

The torque step at 8ms occurs within a single 100μs cycle. The stator flux magnitude is not as sensitive to the applied voltage and there is negligible change over the 100μs while the torque is being driven to the new reference value.

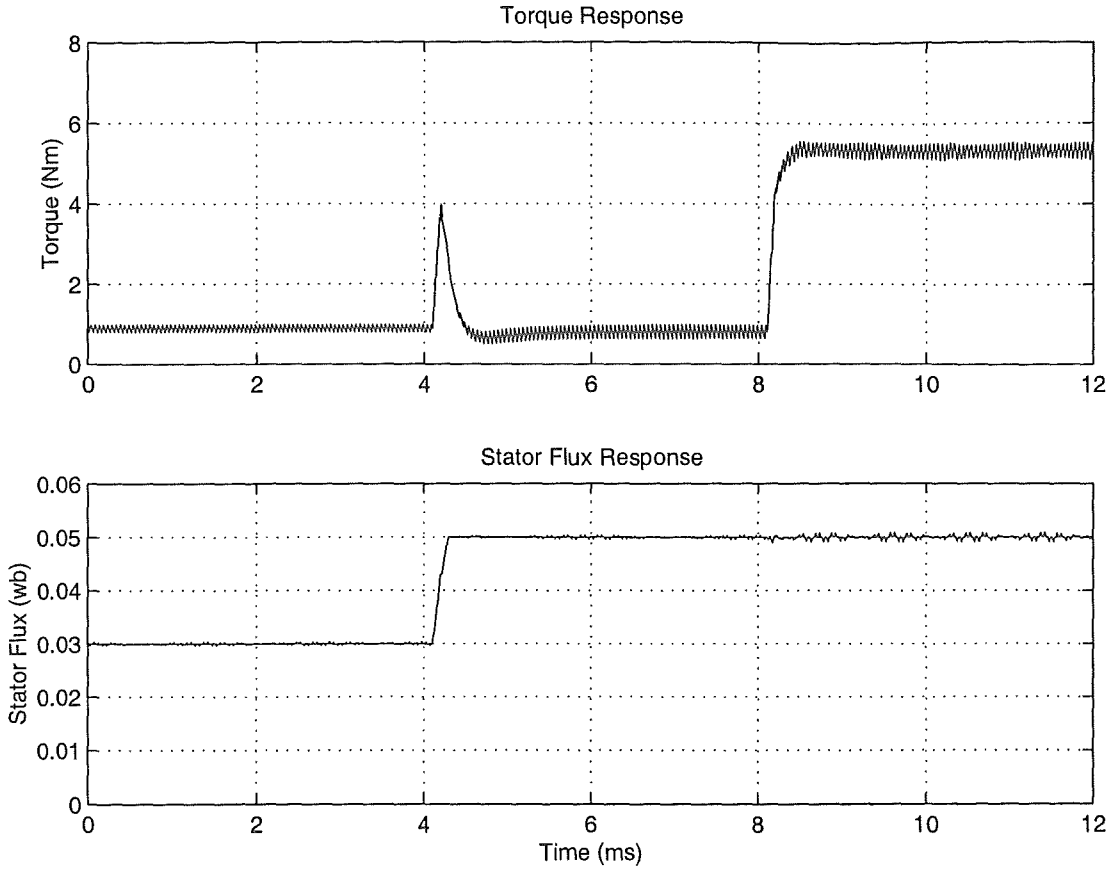


Figure 5.8: Transient performance of DTC using space vector modulation

5.4 Comparison of the torque control schemes

Induction motor torque control has traditionally been achieved using field-oriented control. This is a relatively simple but effective control scheme. Its disadvantages are that it requires an encoder to measure rotor speed, it's sensitive to parameter detuning, and the PI controller that regulates the stator current limits the transient response.

The direct torque control scheme (Section 2.3.2) was later introduced to control the torque and flux directly based on the instantaneous errors in the torque and flux. Although the only motor parameter required by the basic DTC scheme is stator resistance, it has poor performance at low speed due to problems with the stator flux estimation. It

does eliminate the encoder required to measure rotor speed and has significantly higher dynamic performance due to the lack of the PI controller. The hysteresis-based switching of the inverter results in high torque ripple and current distortion. It also results in a variable switching frequency and therefore produces a wide spectrum of harmonics. Space vector modulation at the same average switching frequency results in significantly lower torque ripple and current distortion.

Minimal torque ripple DTC as been published recently [Kang and Sul, 1999] as a technique to minimise the torque ripple produced by classical DTC. This was originally published as a technique to directly control a two-level inverter. Section 2.3.3 presented an extension to this technique to enable it to control a three-level inverter. This extension unfortunately ignores the six inner voltage vectors that the three-level inverter can produce. As a result, the three-level minimal torque ripple DTC scheme described in Section 2.3.3 produces more torque ripple than classical DTC under some operation conditions. The results of Section 5.3.3 show minimal torque ripple DTC still preserves the fast transient response of classical DTC.

DTC using space vector modulation combines the best features of field-oriented control and direct torque control. Like the classical direct torque control scheme, no encoder is required and the lack of a PI current controller results in a high dynamic performance. The disadvantages of classical DTC are removed by the use of space vector modulation that makes effective use of the inverter, resulting in low torque ripple and current distortion.

The disadvantage of DTC using space vector modulation is its high complexity. As Section 3.2.3 showed, the process used to predict the required voltage is computationally intensive. While it is complex compared to field oriented control and classical direct torque control, it can be implemented on what is currently a mid-range digital signal processor, as described in Section 7.3 on the software implementation.

Table 5.1 presents a summary of the advantages and disadvantages of the four torque control schemes that have been considered in this thesis. Classical DTC and minimal torque ripple DTC were determined to be unsuitable for this application due to their high current distortion when compared to the other two control schemes. This high current

distortion would probably cause increased losses in the motor and therefore reduce the overall system efficiency.

DTC using space vector modulation showed a significantly improved transient response compared to field oriented control. Although, field oriented control can achieve relatively quick changes in torque and flux in this application due to the low inductance of the high-speed motor that is used. The simulated 1.5ms response time for a 5Nm step change in torque is more than adequate for this application. Field oriented control would be been suitable for this application, except DTC using space vector modulation is instead chosen as it is a novel alternative and has some additional advantages. These include its improved transient response and the ability to control the torque without requiring an incremental encoder on the motor.

Table 5.1: Advantages and disadvantages of the simulated control schemes

	Advantages	Disadvantages
Field Oriented Control	• Relatively simple high performance control scheme • A proven technique that has been used for some time	• Relatively low dynamic performance due to the PI current regulator • Parameter detuning causes high torque and flux magnitude errors.
Direct Torque Control	• Fast torque response • Relatively simple • No speed or position encoder is required	• High current distortion • High torque ripple • Switching frequency changes with motor speed • Poor low speed performance
Minimal Torque Ripple DTC	• Fast torque response • Low steady-state torque ripple under some operating conditions	• Stator flux ripple and current distortion are higher than for classical DTC
DTC using SVM	• Fast torque response that is equivalent to classical DTC • Low steady state torque ripple and current distortion that is equivalent to field oriented control • Fixed switching frequency • No speed or position encoder is required if the voltage model stator flux observer is used	• The control algorithm is relatively complex compared to the other control schemes. It requires a relatively fast processor to implement at the desired 10kHz switching frequency.

5.5 Summary

This chapter has presented the results of computer simulations that show the steady state and transient performance of four different torque control schemes. Section 5.4 presented a comparison between the four schemes that summarised the simulation results and showed the advantages and disadvantages of each scheme. Based on the results of these

simulations, direct torque control using space vector modulation was chosen as the torque control scheme for this application. This control scheme combines the good steady-state performance of field-oriented control and the fast response of direct torque control.

Direct torque control using space vector modulation is implemented and validated experimentally. Chapters 6 and 7 described the hardware and software used to implement the controller and chapter 8 presents the experimental results obtained. The experimental results of chapter 8 can be compared to the simulation results for DTC using space vector modulation that were presented in this chapter.

6. Hardware Implementation

The direct torque control using space vector modulation control scheme that was simulated in the previous chapter is implemented digitally. The induction motor stator currents and DC bus voltage are digitised and processed by the control algorithms that are implemented using software running on a microprocessor. The complete implementation needed to be capable of stand-alone operation within the electric vehicle. This chapter describes the implementation of the analogue and digital hardware required and the following chapter describes the software created to implement the torque controller.

6.1 Hardware Overview

To enable implementation of the torque controller, the following list of requirements for the system was developed.

1. Measurement of phase currents, DC bus voltage, torque demand, motor and inverter temperature, and motor speed.
2. Analogue to Digital Conversion (ADC) of all analogue measurements
3. A microprocessor to enable a digital implementation of the control algorithms
4. Digital to Analogue Conversion (DAC) to enable calculated data to be viewed in real-time on an oscilloscope
5. Digital interface to the 12 inverter gate drive circuits of the three-level inverter
6. Non-volatile storage of configuration parameters
7. Interface to a Personal Computer (PC) for monitoring and control

The basis of the torque controller is a digital signal processor (DSP) and a programmable logic device. A DSP is a type of microprocessor that has been specially designed to implement the algorithms required to process signals digitally. These algorithms are characterized by repetitive mathematical computation at high-speed, similar to this application where the mathematically intensive torque control and space vector modulation algorithms must be run within every 100μs switching period. Using a programmable logic device enables the implementation of features in hardware that would otherwise need to be performed by the DSP or implemented using discrete digital

logic. Implementing these features in logic frees more of the DSPs time for the mathematical computation needed for the torque control algorithms.

The Texas Instruments TMS320 family of DSPs and the Xilinx range of Field Programmable Gate Arrays (FPGAs) were used since the Department of Electrical and Electronic Engineering at the University of Canterbury has the required development tools and experience with these devices. From the TMS320 range of DSPs, the TMS320C3x family was chosen since it had the required level of performance and a floating-point processor would simplify software development.

Considerable time was spent searching for a suitable development board that contained a C3x DSP and Xilinx FGPA, which could be interfaced to other custom-built hardware, and could operate stand-alone in the electric vehicle. Development boards were readily available for each device, but no suitable development system that contained both devices could be found. Rather than making a compromise with the features available on pre-build development boards, it was decided that a more optimal solution would be to design and build all the necessary hardware with the specific features and specifications that are required.

With the aim of increasing the reusability of the hardware, a modular approach to the design was followed rather than developing a single PCB containing all the features required for this project. The hardware design is based on a base microprocessor board with stackable expansion boards connected via a common bus, similar to the PC/104 bus system used in some industrial PC systems. The hardware is split into the following modules;

1. A base 'Processor Board' containing the DSP and FPGA
2. An 'Analogue Expansion Board' containing ADC and DAC hardware that connects to the processor board
3. 'Analogue Signal Conditioning Boards' to convert the currents and voltages to the signal levels required by the Analogue Expansion Board

Due to complexity of the PCBs required, a commercial PCB prototyping service was used to manufacture four layer PCBs. The sizes of the two main boards were constrained by

the available area on the prototype PCB panel and the physical space needed by I/O connectors. Figure 6.1 shows the mechanical arrangement of the PCBs. The two analogue signal conditioning boards plug into the top of the analogue expansion board via header sockets. The analogue expansion board then plugs into the top of the processor board via the expansion bus connector. The entire stack of PCBs is held together using M3 bolts and nylon spacers. Care was taken when designing the PCBs to ensure no obstructions between components or connectors would occur when they were fitted together. A photograph of the complete assembled PCBs is shown in Figure 6.2.

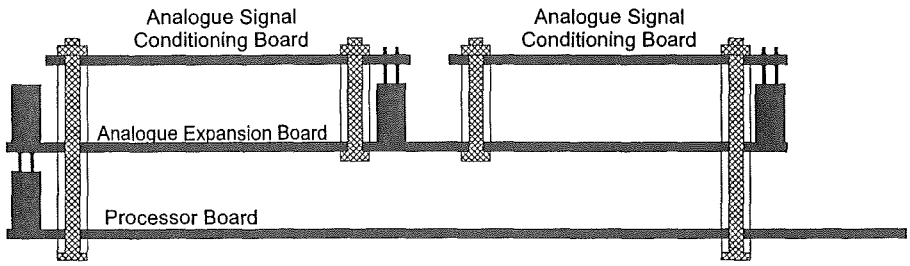


Figure 6.1: Mechanical PCB Arrangement

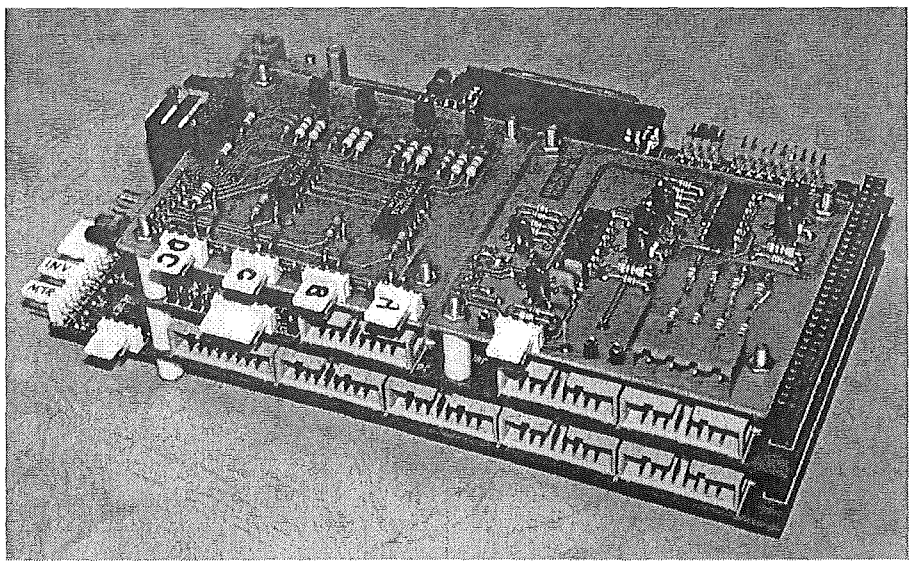


Figure 6.2: Photograph of the assembled PCBs

The individual PCBs developed are explained in more detail in the following sections.

6.2 Processor Board

A diagram of the processor board is shown in Figure 6.3. This board forms the base of the system to which other boards can be attached via its expansion bus. There are five major components to the processor board shown in Figure 6.3;

1. An Expansion Bus to allow for the connection of various expansion boards.
2. A Xilinx XC4020XLA FPGA which interfaces to the DSP and implements all of the logic required
3. A Texas Instruments TMS320VC33 DSP that runs the torque control algorithms
4. An 8-bit PIC16F877 microcontroller which performs simple initialisation and control functions for the processor board
5. Serial Flash memory for the storage of the FPGA configuration, DSP program code, and software configuration information

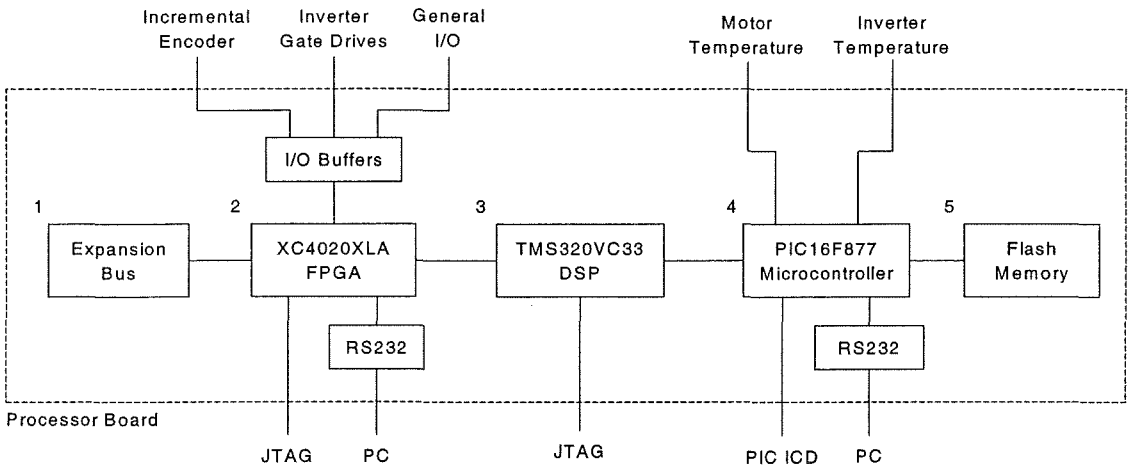


Figure 6.3: Processor Board Block Diagram

A photograph of the processor board is shown in Figure 6.4. The general layout of the components follows the same arrangement as the block diagram of Figure 6.3 in order to keep the PCB routing as short and as simple as possible. The permanent I/O connections to external hardware are at the top and right of the board while the temporary connections to the PC and the status LEDs are at the bottom. This allows the PCB to be mounted edgeways with the PC connections and status LEDs at the front. The complete schematic diagrams for the processor board are included in Appendix B-1.

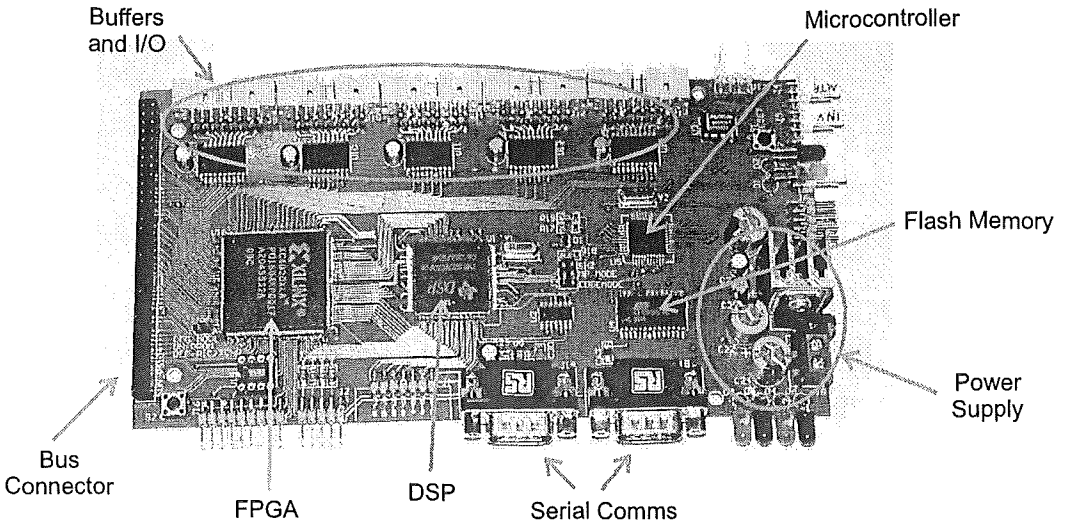


Figure 6.4: Photograph of the processor board

The major components of the processor board are described in more detail in the following sections.

6.2.1 PIC Microcontroller

An 8-bit PIC16F877 microcontroller from Microchip Technology Inc. performs the general initialisation and control functions required by the processor board. The PIC16F877 is currently one of the newest devices in the mid-range PIC16 series. It is flash memory based and supports the PIC In-Circuit Debugger (ICD) interface making it possible to run and debug code while the microcontroller is mounted on the PCB. Connected to the PIC is an Atmel AT45D021 two-megabit serial flash memory chip. This is used to hold the FPGA configuration, DSP program code, and software configuration. When the board first powers up or the PIC is manually reset, it performs the following tasks.

1. It configures the FPGA using data stored in flash memory
2. Releases the DSP reset line
3. Boot-loads the DSP via its Serial Peripheral Interface (SPI) port using program code stored in flash memory
4. Transfers all DSP software configuration parameters (such as induction motor parameters) from the flash memory to the DSP.

Once the processor board is initialised and the DSP is running, the PIC periodically measures the inverter and motor temperature using LM35 temperature sensor ICs and sends this information to the DSP. It also communicates using RS232 serial communications with a host PC running the Motor Manager software (see Section 7.2). This software can modify the data stored in the flash memory, request data from the DSP, or modify the DSP configuration parameters. It was originally intended that a simple display unit would be connected to the PICs RS232 port. This would enable the user to view data and modify the configuration without requiring a PC. However, this was not fully implemented due to time constraints.

6.2.2 Digital Signal Processor

A TMS320VC33 DSP from Texas Instruments is used to implement all the software required to achieve torque control. This DSP is based on the TMS320C31 and is currently the newest device in the C3x family that was first developed in 1988. Compared to previous devices in the C3x family of DSPs, it has the following advantages.

1. A 1.8V processor core and 3.3V I/O give it a very low power consumption
2. 1.1 megabits (34k words) of on-board RAM
3. JTAG interface instead of the TI propriety MPSD port
4. 13ns instruction cycle time for the TMS320VC33-150 compared to a minimum of 25ns for the previous fastest device, the TMS320C31-80
5. Fabricated using modern 0.18μm processing technology enabling more features and higher clock speeds at a lower cost

More details on the TMS320VC33 are included in Appendix D. The most significant advantage of the TMS320VC33 compared to previous C3x devices is the large amount of on-board RAM. This eliminated the need for external high-speed SRAM in this application. A 12MHz quartz crystal is used with the DSP's internal phase looped loop (PLL) to obtain 60MHz H1 and H3 internal clock signals. This results in an instruction cycle time of 17ns or 60MIPs.

The DSP is connected to the PIC microcontroller via its SPI port and this is used to boot-load the DSP at start-up and to transfer configuration and performance data to the PIC. The DSP is dedicated to running the main control algorithms with only a small amount of processing overhead required to service the communication to the PIC microcontroller. The DSP is also connected to the FPGA via its external bus and through the FPGA it can quickly access all the remaining hardware in the system.

6.2.3 FPGA

All of the digital logic on the processor board, with the exception of the buffers, is implemented using a single Xilinx XC4020XLA FPGA. More details on the XC4020XLA are included in Appendix D. The FPGA is connected to the DSP's external bus, the expansion bus, forty buffered I/O lines, and an RS232 transceiver.

The FPGA is interfaced to the DSP using 16 data lines and 8 address lines. It also uses the 60MHz H1 clock from the DSP, which is internally divided down to the 30MHz and 2MHz clocks that are used by the internal logic of the FGPA. The DSP is able to access the expansion bus through the FPGA. This expansion bus is physically based on a 68-way stackable PC/104 style bus connector. Expansion boards connected to the bus can access 36 of the FPGA I/O lines and the power supplies (+3.3V, +5V, +12V) of the processor board.

There are a total of 40 I/O lines from the FPGA that are connected to external I/O connectors via 74HCT244 octal buffers. These buffers serve two purposes, they convert the 3.3V I/O from the FPGA to true 5V CMOS logic levels and provide limited isolation between the FPGA and external hardware. In the event of an external fault damaging the processor board, it would be easier to replace a 20 pin 74HCT244 rather than the 160 pin FPGA. There are twenty outputs and twenty inputs configured as five groups of four outputs and four inputs. Each group of I/O lines go to a 16 way IDC connector along the top edge of the PCB as can be seen in Figure 6.4.

The FPGA configuration was created using a mixture of schematic capture and VHDL code with the Foundation Series software from Xilinx. VHDL is a formal hardware description language that the Foundation Series software can synthesise logic from.

Schematic diagrams are effective for depicting the high level structure of the logic and showing how the various parts are interconnected. For most low-level logic, schematic diagrams become very complex and difficult to understand. It is in these situations, and for representing finite state machines, that a text based VHDL behavioural implementation is significantly easier to understand.

The FPGA can be configured to perform any digital logic function that is required. Figure 6.5 shows a block diagram of the logic that was implemented for this application. Each of the blocks represented in Figure 6.5 are described in more detail in the following sections, while the schematics are included in Appendix C-1.

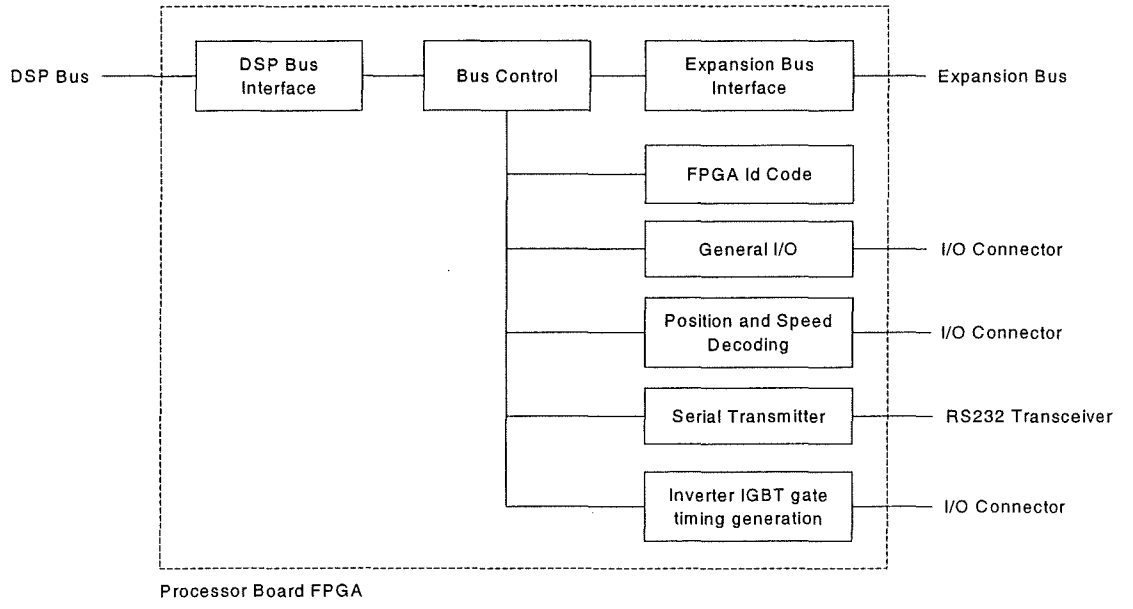


Figure 6.5: Diagram of the processor board FPGA logic

6.2.3.1 DSP Bus Interface

To simplify address decoding, the TMS320VC33 DSP decodes the upper two address lines and combines these with the strobe signal to generate four page strobes ($\overline{PAGE0}$ to $\overline{PAGE3}$). $\overline{PAGE1}$ is used to access the FPGA on the processor board while $\overline{PAGE3}$ is used to access devices on the expansion bus. The FPGA on the processor board is configured as a group of registers that map into the DSP's memory space in the address range of 0x400000 to 0x40001F. The DSP can control the hardware by reading from and writing to data values in this address range.

6.2.3.2 General I/O

There are five inputs and eight outputs on the external I/O connectors that were not needed for any dedicated logic function. These I/O are arranged into two registers such that DSP can directly read the inputs and control the outputs. A simple PCB was created that connects to one of the IDC sockets. This contains Start, Stop, Trip, and Trip Reset buttons and Run, Warn, and Tripped status LEDs that the DSP can access.

6.2.3.3 Rotor Position and Speed Decoding

Rotor position and speed information is obtained using an incremental encoder coupled to the rotor shaft. The incremental encoder produces two digital pulse streams that are in quadrature, with the relative phase and the frequency of these pulse streams used to determine the rotor direction, position, and speed. The two digital outputs of the encoder are sampled at 30MHz and passed through an edge detector. With each encoder slot generating a rising and falling edge on each output channel, a total of four pulse edges are generated for each encoder slot.

The rotor position is measured by using the detected pulse edges to clock a counter that counts up or down based on the rotor direction, which is determined by the relative phase of the two encoder outputs. When the counter reaches the pre-configured number of pulses per revolution while counting up, it wraps around back to zero. When it reaches zero while counting down it wraps back around to the configured maximum pulse count. Using the current pulse count and the number of pulses per revolution, the angle relative to an arbitrary starting position can be calculated.

The conventional method of determining rotor speed is to count the number of pulses that occur within a fixed time period. The time period must be long enough such that sufficient pulses occur to obtain an accurate measurement. A disadvantage of this technique is the long delay between successive updates to the speed measurement. A faster update rate, especially at high speed, can be obtained by measuring the time between encoder pulses. This requires a high timer resolution, but is not a problem in this application due to the 33ns resolution that can be obtained with the available 30MHz clock. In practice, the timing is performed over multiple pulses to obtain an average per pulse. This reduces the effect of errors in slot and sensor position alignment.

The timing and pulse counting required to implement rotor speed measurement is implemented entirely by the FPGA. The desired total pulse count is loaded into a register by the DSP. This action resets the pulse counter and starts a timer. The pulse counter increments on each pulse edge of the incremental encoder. When the counter reaches the configured total pulse count the timer stops. The current timer value is then stored in a holding register, the timer and pulse counter are reset, and a ready flag is set. The DSP periodically polls the ready flag and if set reads the timer holding register, which in turn clears the ready flag. This allows the FPGA to start taking a new speed measurement while the DSP is still reading the previous one. The DSP can then calculate the rotational speed based on the measured time for a known number of pulse edges and knowing the number of pulses per revolution.

6.2.3.4 Asynchronous Serial Transmitter

The DSP is able to send data at high baud rates to a PC for analysis using an asynchronous serial transmitter build into the FPGA and an external RS232 transceiver. The asynchronous serial transmitter is effectively a parallel to serial converter that creates serial data frames by adding start and stop bits. The serial data rate is configurable to allow for different baud rates to the PC to be used. A 16-byte First In First Out (FIFO) queue is also used to maintain high data throughput with minimal DSP overhead. Data reception was not required and was therefore not implemented. Low speed communication to and from the PC is performed using the DSPs SPI interface via the PIC microcontroller.

6.2.3.5 Inverter Gate Signal Timing Generation

The most important function of the FPGA is the generation of the timing for the inverter IGBT gate signals. The inverter gate-signal timing module has the following features.

- Generation of IGBT gate signal timing with minimal DSP execution time required
- Insertion of a dead-time between IGBT commutations
- Protection against illegal switch combinations that could cause short-circuits within an inverter phase leg. For a three-level inverter there are 16 possible IGBT combinations per phase but only four valid ones.

- Deactivation of all IGBTs in the event of a fault (IGBT desaturation, over-current, or external trip signal)

Figure 6.6 shows a block diagram of the IGBT gate-signal timing module. It consists of six stages, which are described in the remainder of this section.

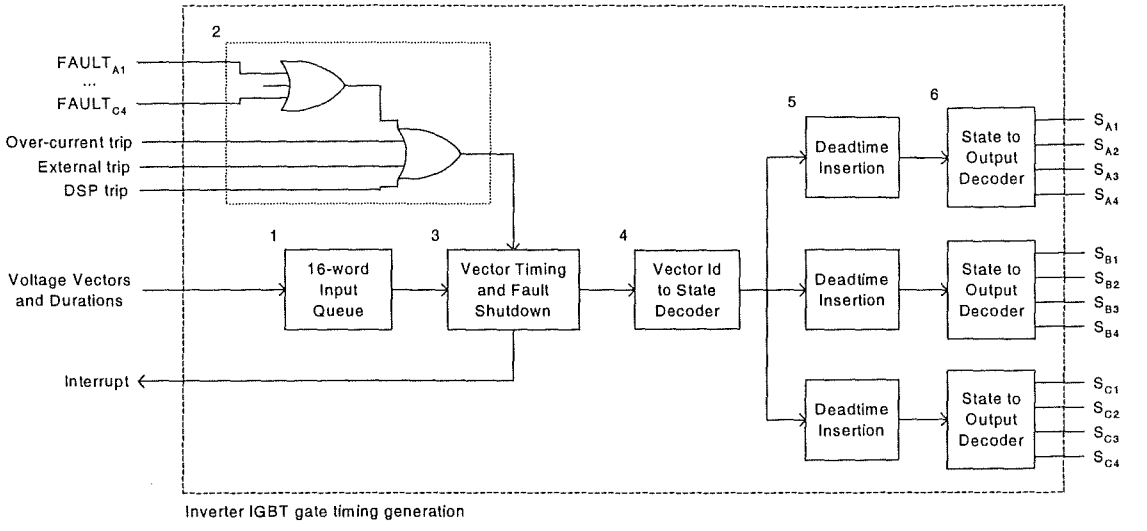


Figure 6.6: Block diagram of the IGBT gate signal timing module

1. The DSPs space vector modulation algorithm calculates the six voltage vectors and corresponding on-durations required for the next switching cycle. Each vector is encoded by the DSP as follows;
   - 5-bit voltage vector id
   - 1-bit interrupt state
   - 16-bit count for the on-duration timer

   The DSP writes the vectors into a 22-bit by 16-word deep FIFO queue. Using a FIFO allows all the information for the next switching cycle to be transferred to the FPGA and eliminates the need to interrupt the DSP for each switching instance. Also, if the DSP fails for some result and stops writing vectors into the FIFO, the FIFO eventually empties, and all the IGBTs are turned off. In the event of a DSP software fault, this prevents a single voltage vector from being held being permanently and effectively producing a DC output voltage.

2. If any fault condition occurs the module enters a 'Tripped' state in which all of the IGBTs are turned off. A trip condition occurs when any of the following signals become active;
   - Any of the 12 IGBT desaturation fault signals from the gate drives
   - Over-current detection for each current input
   - An external trip signal
   - An internal trip signal from the DSP

   The trip state can only be cleared by the DSP specifically resetting the trip condition.

3. A voltage vector timing stage fetches and decodes vectors from the FIFO.
   - The on-duration count is used by a 16-bit timer is used to determine how long each vector should remain active for. Running from a 30MHz clock, the timer has a resolution of 33ns and a maximum time period of 2.18ms, which allows switching frequencies down to 460Hz. When the timer expires the next vector is fetched from the queue.
   - The 5-bit voltage vector id is passed to the next stage for decoding
   - The 1-bit interrupt state is used to drive the interrupt line of the DSP. A DSP interrupt is generated when a vector is fetched from the queue that has this bit set. It used to inform the DSP that it should run the space vector modulation algorithm again for the next switching cycle. This interrupt ensures the DSP and the inverter timing generation remain synchronised.

4. A lookup table is used to decode the voltage vector id into a 2-bit value for the output state of each phase. The four possible output states for are off, positive, zero, and negative.

5. A dead-time insertion stage for each phase detects changes in the output state of the previous stage. When a transition from one state to another is detected, the output of that phase changes to the 'Off' state and a configurable 8-bit timer is started. The output changes to the new state when the timer expires.

6. The last stage is a lookup table that decodes the 2-bit output state into the gate signals for the IGBTs that need to be on to achieve that state.

6.2.3.6 Expansion Bus

Thirty-six of the FPGA's I/O lines are exposed via the expansion bus. To enable the common interconnection of expansion boards, these I/O lines have been allocated for 8-bit addresses, 16-bit data, two interrupts, six general I/O, a 2MHz clock and various control lines. One of the general I/O lines has been allocated for the over-current trip signal to the inverter IGBT timing module, but the other five were not used for this application.

The 8-bit address from the DSP is passed directly to the expansion bus while the buffering of the 16-bit DSP data lines is controlled by the $R/\overline{W}$ line and the $\overline{PAGE3}$ strobe signal (active for addresses 0xC00000 to 0xFFFFFF). The expansion bus data lines are only driven during a write to the expansion bus and they are gated onto the DSP bus during a read. The expansion bus allows the DSP to quickly access and control other PCBs. For this application an analogue expansion board has been created.

6.3 Analogue Expansion Board

The analogue expansion board is designed to connect to the expansion bus of the processor board and provides the following features.

1. Analogue to Digital Conversion (ADC) of measured signals such as phase currents and DC voltages
2. Digital to Analogue Conversion (DAC) to allow real time observation of calculated values on an oscilloscope
3. An Enhanced Parallel Port (EPP) interface for high speed communication with a PC
4. General Digital I/O

A block diagram of the analogue expansion board is shown in Figure 6.7. A Xilinx Spartan XCS10XL FPGA provides the logic required to interface this board to the

expansion bus. An introduction to the Spartan devices from the XCS10XL datasheet can be found in Appendix D. The FPGA implements the parallel port interface, the registers for the general I/O lines, and the Finite State Machines (FSMs) that control the A/D and D/A conversion processes. The schematics for the FPGA logic are included in Appendix C-2.

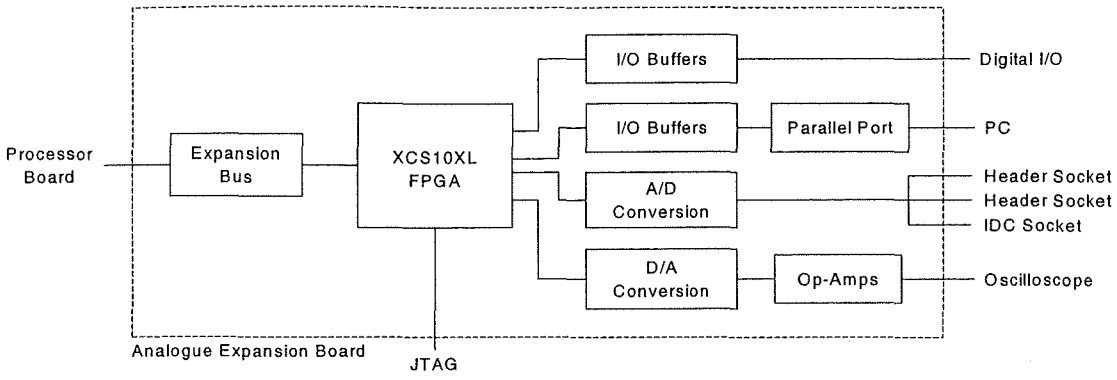


Figure 6.7: Analogue Expansion Board Block Diagram

Figure 6.8 shows a photograph of the analogue expansion board. It follows the same layout as the processor board with permanent external connections along the top and the parallel port PC connection and status LEDs along the bottom. Similar to the processor board, it has two 16-way IDC connectors that provide eight outputs and eight inputs via 74HCT244 octal buffers. On the face of the board are two header sockets that allow PCBs to be added that provide the signal conditioning required by the ADC. While +3.3V, +5V and +12V supplies are provided via the expansion bus, ±15V is required by some of the analogue devices and to generate the ±5V required by the ADC. This is either supplied by DC/DC converter, or as in this case, from an external supply.

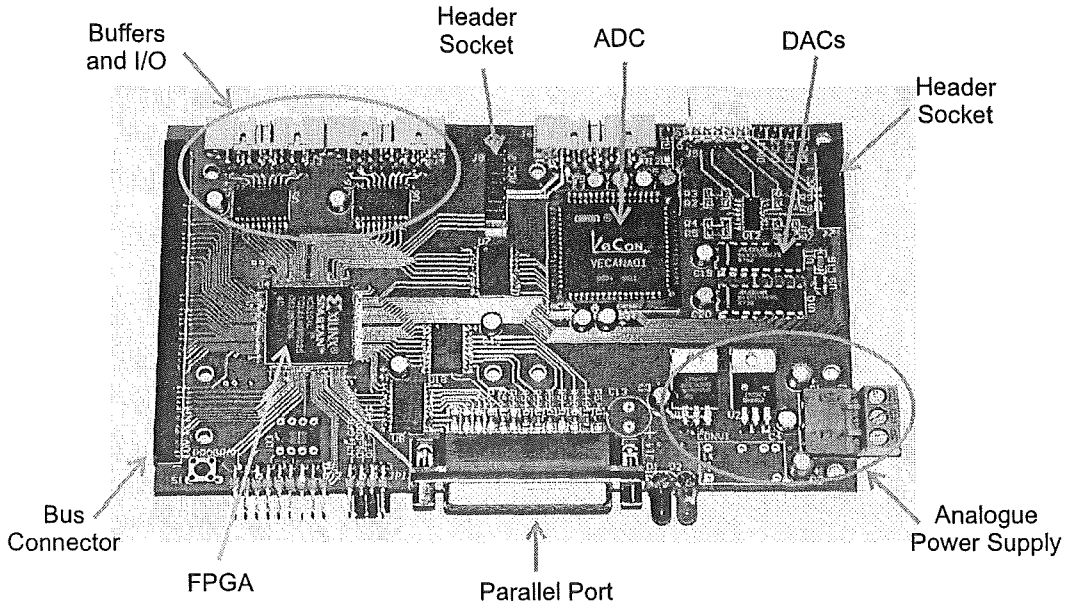


Figure 6.8: Photograph of the Analogue Expansion Board

The following sections describe the parallel port interface, and the A/D and D/A converters in more detail. The full schematics for the analogue expansion board are included in Appendix B-2.

6.3.1 Parallel Port Interface

An interface to the parallel port of a PC was included to enable high-speed data transfer. This port operates in Enhanced Parallel Port (EPP) mode where the handshaking signals are automatically generated by the FPGA. This is in comparison to the Simple Parallel Port (SPP) or Centronix modes where they are generated by software. In EPP mode data transfer rates of up to 500 kB/sec are possible compared to a serial port operating at 115200bps that has a data transfer rate of about 11.5 kB/sec. This interface was never utilised in this application, but was included in case high-speed data transfer to a PC was required.

6.3.2 Digital to Analogue Conversion

During operation, the DSP measures and calculates a large amount of information every 100μs. Two DACs are used to convert the digital data from the DSP into analogue voltages that can be observed in real-time using an oscilloscope. 10-bit MAX5159 and 12-bit MAX5155 dual serial DACs from Maxim are used. This gives a total of four

channels that are amplified using an LM324 op amp to a 0 to 10V output. At the end of every processing cycle the DSP updates the data values stored in the FPGA. The FPGA then clocks out the new data values to both DACs simultaneously.

6.3.3 Analogue to Digital Conversion

Analogue to digital conversion is achieved using an ADS7833[1] A/D converter from Burr-Brown. The ADS7833 contains three separate analogue-to-digital converters (ADCs), each of which has a three-way input multiplexer. More details on the ADS7833 can be found in Appendix D. When a conversion is initiated by the DSP, the FPGA clocks out the control information for the next conversion while simultaneously clocking the current conversion results for all three ADCs. All the differential analogue inputs of the ADS7833 are brought out to connectors, two header sockets on the face of the board and an IDC connector on its edge. The multiplexer of the ADC is used to select between each socket.

Each socket has the differential inputs to each of the three ADCs and a ±15V analogue supply. The two header sockets on the face of the board also each have four digital I/O lines to the FPGA and a +5V digital supply. These header sockets are used to connect analogue signal conditioning boards that convert the measured signal to the ±2.5V input signal required by the ADC. In this way the ADC is not limited to the conversion of any particular signal since interchangeable boards are used to perform any filtering or amplification that is required.

6.4 Analogue Signal Conditioning Boards

As described in Section 6.3, the Analogue Expansion board has two header sockets on the face of the board that provide analogue inputs, digital I/O, and power supplies for the connection of analogue signal conditioning boards. Two such boards have been created for this application, one for the measurement of the motor stator current and another for the measurement of the DC bus voltage and torque demand. These two boards are described in the following two sections.

[1] The ADS7833 is now obsolete and has been replaced with the VECANA01 which is pin compatible

6.4.1 Current Measurement

For measurement of the stator current, LEM LA-100S current transducers were used. These are a hall-effect based device that have a –1dB frequency response from DC to 100kHz. They use a ±15V supply voltage and produce a 1:2000 ratio output current from the measured current. On the current measurement board, the current output of the LEM sensor is converted to a voltage using a 33Ω resistor, giving a current range of ±150A. The current measurement board has four channels allowing for the measurement of the three-phase currents and the DC link current (if required).

The phase current measurement board also provides over-current protection. The voltages created from each current input are fed into window comparators created using LM339 quad comparators. If the current input exceeds the configured range, the output for that comparator becomes active. The four comparator outputs are fed via the digital I/O lines in the header socket to the FPGA were they are latched and an over-current trip signal is sent to the inverter gate timing logic on the processor board (See Section 6.2.3.5).

Figure 6.9 shows a photograph of the current measurement board. The schematic diagram is included in Appendix B-3.

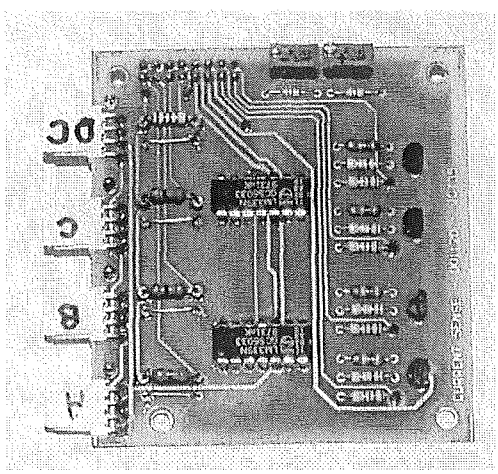


Figure 6.9: Photograph of the Current Measurement Board

6.4.2 DC Bus Voltage Measurement

The DC bus voltage measurement board contains two separate differential amplifiers. One differential amplifier measures the upper half of the DC bus while the other measures the lower half. A division ratio of 67:1 is used to enable measurement of voltages up to 166V, which corresponds to a total DC bus voltage of 332v. RC filters are used to remove the considerable amount of inverter switching noise that is present on the incoming signals. The third ADC channel available on this board is used measure the torque demand. A socket is provided for the connection of a potentiometer between the ±15V analogue supply lines.

Figure 6.10 shows a photograph of the voltage measurement board. The schematic diagram is included in Appendix B-3.

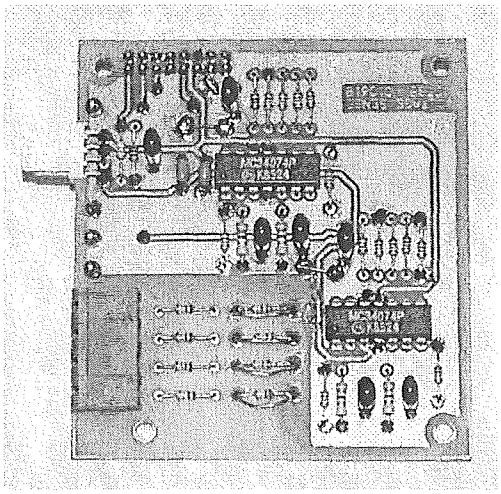


Figure 6.10: Photograph of the Voltage Measurement Board

6.5 Summary

The hardware that was developed to implement the torque controller consists of a base digital processor board, an analogue expansion board, and two analogue signal conditioning boards. The two signal conditioning boards convert the phase currents and DC bus voltages to a ±2.5V input for the ADC. The analogue expansion board contains the ADC for digitising the analogue signals and DACs to output signals to an oscilloscope. An FPGA is used to interface the ADC and DAC to the expansion bus of the processor board. The processor board contains an 8-bit PIC microcontroller that

controls the board, a 32-bit floating-point DSP to implement the control algorithms, and an FPGA that provides digital logic to implement speed decoding and IGBT timing signal generation.

The following chapter describes the software that runs on the PIC microcontroller to control the processor board and the software for the DSP that implements the torque control algorithms. It also describes two components of software that run on a PC and communicate with the processor board using RS232 serial connections.

7. Software Implementation

As described in the previous chapter, the processor board contains two microprocessors. These are 8-bit PIC microcontroller and a 32-bit floating point DSP. This chapter describes the software implemented for these microprocessors and two associated components of software that run on a PC.

The PIC microcontroller runs the 'System Control' software described in Section 7.1. This initialises the processor board by configuring the FPGA and boot-loading the DSP. It also communicates with the PC based 'Motor Manager' application described in Section 7.2. This software allows a user to upload new FPGA configurations and DSP program code, view the DSP and PIC configuration and data, and generally control the system.

The DSP runs the 'Motor Control' software described in Section 7.3. This samples data using the FPGA and the analogue expansion board and runs all the control algorithms. Data that has been measured and calculated by the DSP is sent to a PC running the 'Motor Monitor' software for analysis. This software, described in Section 7.4, displays the data received and stores it so that it can later be analysed in detail.

Figure 7.1 shows a diagram of how the four software components interact. The two software applications running on the PC communicate with the processor board over two independent RS232 serial connections. On the processor board the PIC and DSP communicate through their SPI ports.

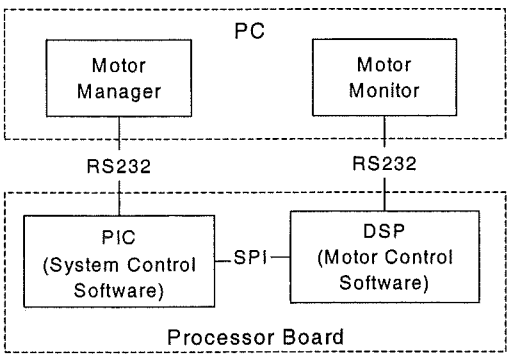


Figure 7.1: Interaction of Software Components

All of the information stored on the processor board is in the form of tables that are accessible to the other software components via the serial connections. There are a total of four tables; PIC Configuration, PIC Data, DSP Configuration, and DSP Data. Configuration tables hold information that defines the operation of the software and can be retrieved and modified by the other software components. Data tables hold acquired and calculated data and are typically read-only to the other software components. All entries in the tables are stored as floating-point values and each software component knows of the arrangement of entries within the tables.

The following sections describe each of the four components of software developed.

7.1 System Control (PIC) Software

The PIC microcontroller initialises and controls the processor board. When the processor board first powers up, it configures the FPGA and boot-loads the DSP using data from flash memory. Once the entire processor board is operational, the PIC communicates with the Motor Manager software running on a PC. The Motor Manager software allows the PC to modify the flash memory, access configuration and performance information, and to send commands in order to control the system.

The following sections describe how the PIC software was developed and the tasks that it performs. Section 7.1.6 describes how it was intended that an external display unit would be connected to the PIC to enable the configuration to be viewed and modified. Although the software for this feature was not fully implemented, a description is included since the inclusion of this feature significantly affected the design of both the hardware and software.

7.1.1 Development Tools

As described in Section 6.2.1, the PIC16F877 microcontroller has an In-Circuit Debugger (ICD) interface. In-circuit debugging is achieved using an ICD interface for the PC and MPLAB software from Microchip Technologies Inc. While the MPLAB software enabled single stepping, breakpoints, and access to internal memory, it was slow to

compile, download, and to respond when using the debugger. An alternative development environment was constructed.

The Hi-Tech 'C' complier, used to build the PIC object code, was integrated into the Microsoft Visual C++ Integrated Development Environment (IDE). This allowed the sophisticated editing features of Visual C++ to be used, while building object code for the PIC. A simple bootloader was loaded into the space within the PIC program memory that is normally reserved for the ICD software. At start-up, the bootloader checks for a serial connection to a corresponding application running on a PC that is used to program the new object code.

Debugging of the software is achieved using TRACE and ASSERT debug statements within the code. These statements send serial data using a general I/O pin and an external RS232 transceiver to terminal software running on a PC. Using this technique it is possible to observe the execution path and internal state of the software. When a fault is discovered the code can be modified, recompiled, and downloaded via the bootloader significantly quicker than using the MPLAB software. This takes about one minute to perform compared to three minutes when using MPLAB.

7.1.2 Flash Memory

A major function of the PIC microcontroller is to initialise the processor board at start-up using data from non-volatile memory. As described in Section 6.2.1, an Atmel AT45D021 flash memory chip is connected to the PIC using general I/O lines. Code written in assembler and optimised for speed is used to transfer serial data to and from the flash memory using these I/O lines. A number of low-level functions are used to implement the command sequences required to read, write and erase the flash memory device. A high-level memory access module uses these functions and the internal buffer of the AT45D021 to provide generic memory read/write access, automatically taking care of accesses that are split across page boundaries.

7.1.3 Serial Communication

Once the processor board has been initialised, most of the PIC's time is spent performing serial communication. It responds to requests from the PC to read, write and erase flash memory, read/write the configuration and data tables, and to accept commands. The PIC also communicates with the DSP to transfer data and configuration information and to pass on commands.

A simple master/slave protocol was defined for all communication. The master device initiates communication by sending a request. The slave device processes the request and sends a response. The format of the request and response packets is shown in Figure 7.2. The start byte is a fixed value that is used for synchronisation. This is followed by the length byte that defines the total number bytes in the packet, a command code that defines the purpose of the request, and any data specific to that command code. At the end of the packet is a modulo-256 checksum taken over all of the preceding bytes in that packet. This is used when the packet is received to ensure that it has not become corrupted. When the slave device sends the response packet, the command code is an echo of the code from the corresponding request. The response also contains a status byte that is used to indicate 'acknowledge' and 'not acknowledge' conditions back to the master device. The exact encoding of the data that these packets carry depends on the specific command code.

Request Format

Start Byte (0x64)	Length 1 byte	Command Code 1 byte	Command Specific Data n bytes	Checksum 1 byte

Response Format

Start Byte (0x64)	Length 1 byte	Command Code 1 byte	Status 1 byte	Command Specific Data n bytes	Checksum 1 byte

Figure 7.2: Protocol Request/Response Format

For PC to PIC communication, command codes were defined for most of the flash memory command sequences. This enabled the Motor Manager software to have full, low level access to the flash memory via its RS232 connection. For PC to PIC and PIC to DSP communication, 'Control Function', 'Check System OK', 'Read Table', and 'Write Table' command codes were defined.

If the PC sends a 'Control Function' or 'Check System OK' request to the PIC, the PIC immediately generates an identical request to the DSP since these commands are only handled by the DSP. When the PIC receives the response, it is passed back to the PC. 'Read Table' and 'Write Table' requests for the PIC configuration and data tables are handled by the PIC, while requests for the DSP configuration and data tables are passed on to the DSP.

7.1.4 FPGA Configuration

The FPGA is configured using data from the flash memory. The Xilinx Foundation Series software generates a '.bit' file that is then formatted into a '.raw' file using a specially written PC application. The '.raw' file is preformatted such that the PIC can transfer the data directly from flash memory to the FPGA without any further processing. The '.raw' file is loaded into the flash memory by the PC. At start-up, the data is read from the flash memory and clocked out to the FPGA using the PICs general I/O lines.

7.1.5 DSP Configuration

The Texas Instruments linker for the C3x DSPs produces a Common Object File Format (COFF) output, however the DSP bootloader requires a specially formatted boot-table that defines blocks of data and their destination address in the DSP. The COFF file from the linker is converted to a boot-table in ASCII hexadecimal (hex) format using the Hex30 conversion utility. The ASCII hex data is then converted into binary hex data that is programmed into the flash memory. At start-up, the data is read from flash memory and transferred directly to the DSP bootloader via the SPI interface.

Once the PIC has boot-loaded the DSP, the PIC starts sending 'Check System OK' requests. If the DSP sends a positive acknowledgement then the PIC assumes that the DSP software is initialised and running. The PIC then sends the entire DSP configuration from flash memory to the DSP using 'Write Table' requests. All of the DSP control software is disabled until it receives the entire configuration table. After the table has been transferred, the PIC keeps sending 'Check System OK' requests once per second. If the DSP does not response to three of these requests, the PIC assumes the DSP has been restarted and it resends the configuration table when the DSP next starts responding.

7.1.6 External Display Unit

Once the FPGA configuration and DSP program code has been loaded into the flash memory, a PC is only needed to view data and modify the configuration. It was intended that a simple external display unit consisting of a two-line 16-character LCD display and five buttons would be connected to the RS232 port of PIC. It would allow the user to view the configuration and data tables, modify the configuration, and send control commands to the DSP.

It was intended that the data and configuration table entries would to be accessed through a system of menus generated on the display. In order to keep the PIC software simple, for each item in the menu system the flash memory would store information on its links to other items, where to retrieve its current data value from, how it was to be displayed, and upper and lower limits for configurable values. The PIC would only need to be programmed with a relatively simple set of rules for processing the menu system data. For example, if the 'Next' button were pressed it would navigate to the next item relative to the current item, lookup the data value from the appropriate table and write it to the display using predefined formatting information. More information on the design of the menu system can be found in Section 7.2.3 where the Motor Manager software is described.

The software necessary for the external display unit was not fully implemented due to time constraints. All of the functionality needed for the development of the torque controller was provided by the Motor Manager application described in Section 7.2.

7.2 Motor Manager Application

The Motor Manager software is a PC based application for the configuration, control, and monitoring of the complete torque control system. The Motor Manager application is used to achieve the following tasks.

- Transfer data to and from the flash memory

- View and modify the PIC and DSP software configuration
- View data acquired or calculated by the PIC and DSP
- Send commands to the torque controller (start/stop/reset)
- Dynamically build the menu system for the external display unit

The Motor Manager application was created using the Visual C++ software development tool from Microsoft. The Microsoft Foundation Classes (MFC) application framework for Visual C++ was used to enable rapid object-oriented software development.

The following sections provide more information on the functions that the Motor Manager application performs and how these are achieved.

7.2.1 Table Management

A major feature of the Motor Manager application is the ability to access the configuration and data tables of the PIC and DSP. To achieve this, it is necessary for the Motor Manager application to know what data is stored in the tables and where. This information is entered using the 'Manage Tables' dialog box shown in Figure 7.3. A symbol and a description are specified for every index used within the four tables. The information entered can be exported as a 'C' header file that consists of `#define` statements for all the symbols. This header file can be used when building the PIC and DSP software to ensure all table definitions are consistent.

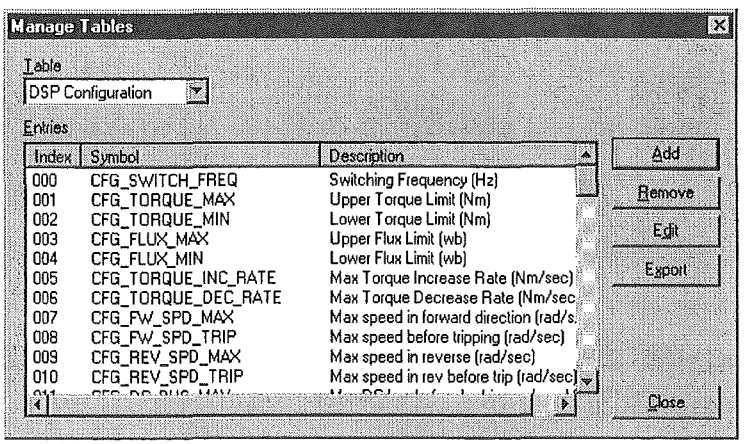


Figure 7.3: 'Manage Tables' Dialog Box

7.2.2 Main Application Window

A screenshot of the main window of the Motor Manager application is shown in Figure 7.4. It is intentionally structured like the Microsoft Windows Explorer and Registry Editor applications to make it intuitive to use. The main window consists of two panes. In the left pane is a hierarchical menu system containing various configuration and data values. In the right pane is a list that displays more information about the items in the currently selected menu.

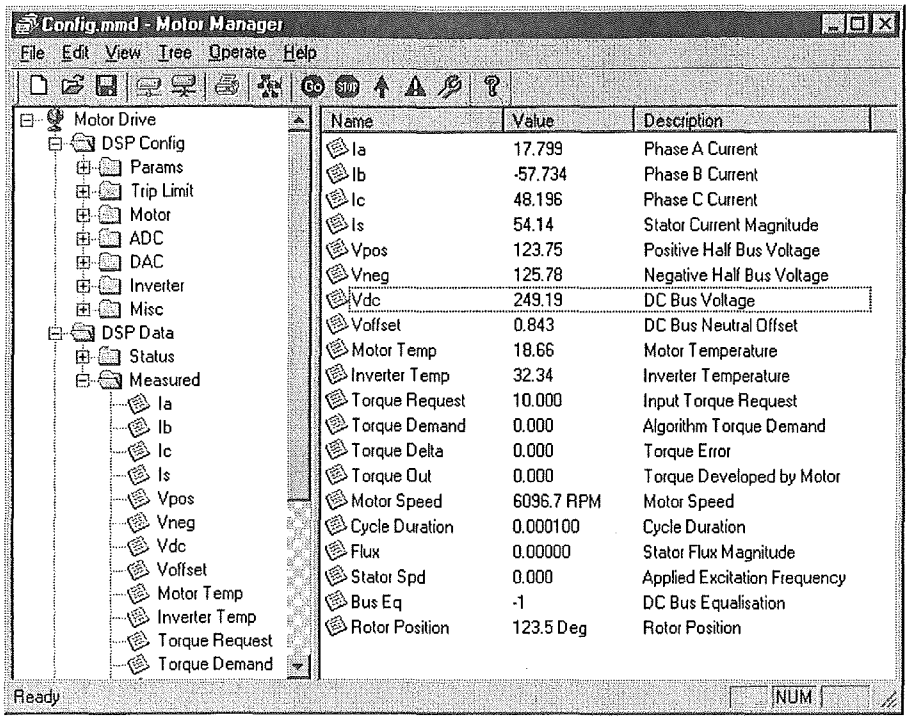


Figure 7.4: Manager Manger application main window screenshot

7.2.3 Menu System Creation

The configuration and data tables of the PIC and DSP are effectively a flat file structure that is convenient for the software to manipulate. The hierarchical menu system implemented by the Motor Manager application provides a way to organise and group the information. The menu system was continuously modified as the development of the system progressed and more items were added to the tables. New menus, Data Views, and Configurable Values could be added at any position in the menu hierarchy. Data Views are read-only values while Configurable Values can be read and written and are stored in flash memory. To insert new Data Views and Configurable Values into the menu system, the dialog boxes described in Sections 7.2.3.1 and 7.2.3.2 were used.

7.2.3.1 Adding Data Views

Figure 7.5 shows a screenshot of the dialog box used to add a new Data View item to the menu system. Using this dialog box the items name, description, the table location to retrieve the data value from, and how to format the data for display are specified.

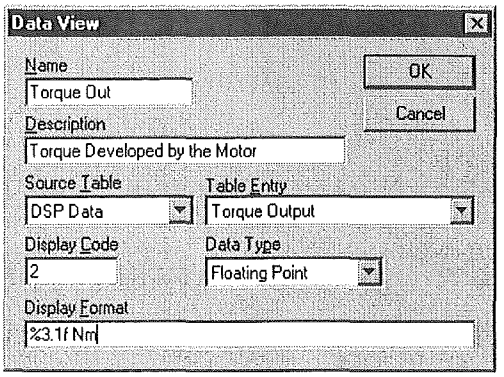


Figure 7.5: 'Data View' Dialog Box

7.2.3.2 Adding Configurable Values

The method for specifying a Configurable Value is similar to adding a Data View. The dialog box used to add new Configurable Values is shown in Figure 7.6. The extra fields on this dialog determine when the value can be modified (when the motor is running, stopped, or anytime), the upper and lower limits, and the step size when increasing or decreasing the value.

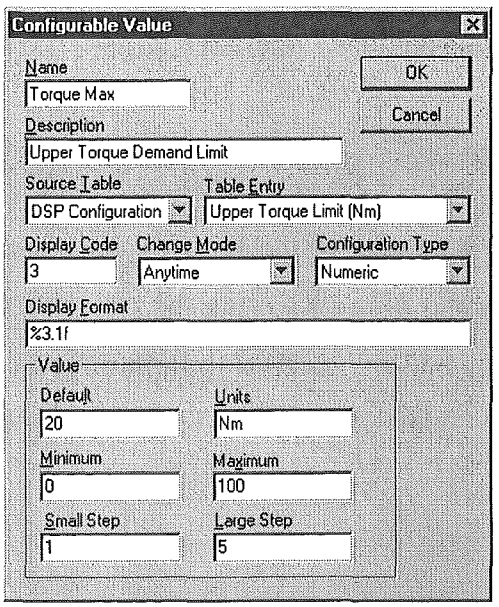


Figure 7.6: 'Configurable Value' Dialog Box

7.2.3.3 Menu System on the External Display Unit

The Motor Manager application was designed to generate the menu system for use on both the PC and the external display unit. The layout of the menu system and all the data entered is written into the flash memory of the processor board. As described in Section 7.1.6, the PIC was designed to interpret this information and re-create the menu system on the external display unit.

7.2.4 Data Display and Modification

When a menu is selected in the left pane of the main application window of Figure 7.4, the right pane is updated to display the information for that menu. For each item, the name, value, and description is listed. The value of each item is updated five times per second by sending a 'Read Table' request to the PIC. When doubling-clicking on configuration items, another dialog box is displayed to allow the value to be modified. When 'Apply' or 'OK' is selected, a 'Write Table' request is sent to update the configuration value.

7.2.5 System Control

It is possible to send commands from the Motor Manager application to the DSP. This includes commands such as 'Start', 'Stop', 'Trip', and 'Trip Reset'. Selecting the appropriate menu item or toolbar button causes the Motor Manager application to send a 'Control Function' request with the necessary function code.

7.2.6 View and Edit Flash Memory

The Motor Manager application is used to upload new data to the flash memory and download existing data. Data can be transferred to and from files using the dialog box shown in Figure 7.7. It can program data from a file into any location in flash memory. It can also read a block of data by specifying a start address, length, and the name of the file to write the data into.

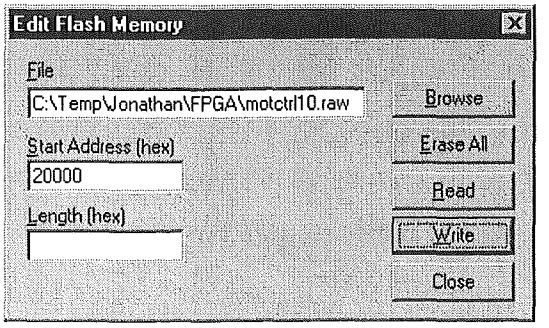


Figure 7.7: 'Edit Flash Memory' Dialog Box

7.3 Motor Control (DSP) Software

The primary purpose of the DSP is to run the control algorithms that calculate the inverter switching signals in order to achieve a desired torque output. The DSP interfaces to the hardware to collect data for control and protection purposes and runs the torque control and the space vector modulation algorithms (described in chapter 3 and chapter 4 respectively). The secondary task of the DSP is communication with the PIC microcontroller. It is through the SPI interface to the PIC that the DSP receives operating commands and all configuration information.

Section 7.3.1 describes the sequence of tasks that the DSP performs. These are divided into data acquisition, outer control loops, direct torque control, space vector modulation, data output, and are described in detail in Sections 7.3.2 to 7.3.6. Section 7.3.7 discusses some issues relating of the performance of the DSP and Section 7.3.8 explains how the DSP control algorithms were simulated using MATLAB and Simulink.

7.3.1 Software Flow

For a switching frequency of 10kHz, each switching cycle lasts 100μs and the processing of all control calculations must be performed within this time period. Instead of starting the processing for the next cycle at the start of the switching cycle, it is started in the exact middle of the switching cycle for two reasons.

1. The calculation delay between sampling currents and applying the calculated voltage is reduced from 100μs to 50μs.

2. There are times when the inverter will switch during the transition from one switching cycle to the next in order to maintain DC bus voltage equalisation. Each time the inverter switches, noise is generated on the LEM sensor output. Starting the processing cycle at the midpoint of the switching cycle avoids sampling the switching noise, since there is not normally any switching at this point.

Starting the processing cycle in the middle of the switching cycle means there is a critical 50μs period in which the currents must be sampled, control algorithms run, and the next set of voltage vectors calculated. Figure 7.8 shows a timeline of the tasks that the DSP performs in each cycle.

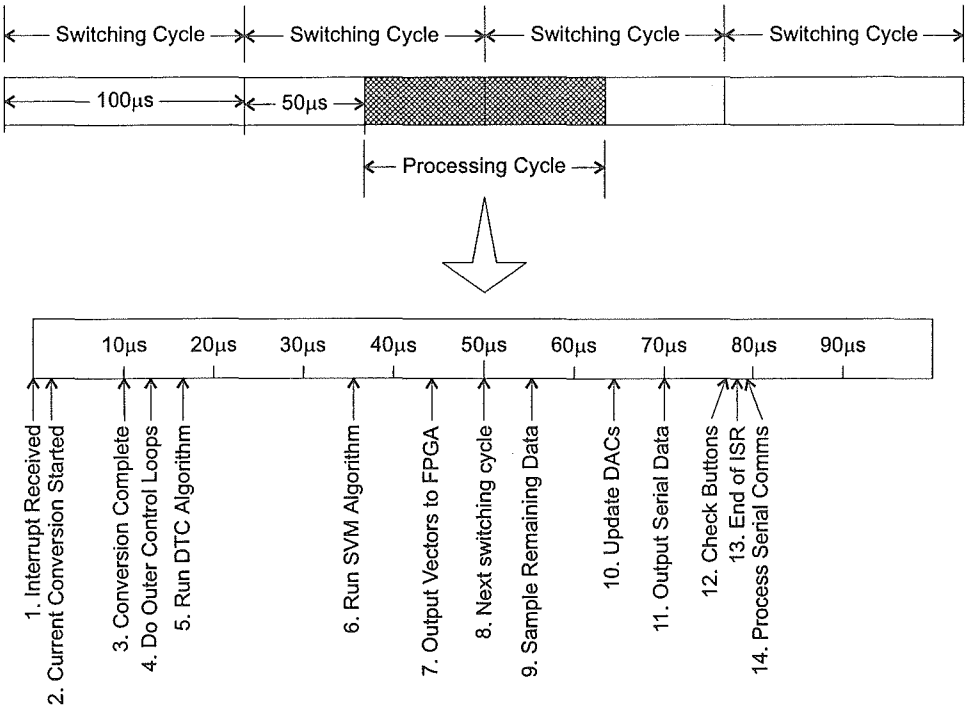


Figure 7.8: DSP Software Timeline

At the midpoint of the switching cycle, the main DSP interrupt service routine (ISR) is run (1) and the motor stator currents are immediately sampled (2) while no inverter switching is occurring. When the conversion is complete (3), the control and SVM algorithms are run (4)-(6) using the newly measured stator current. The voltage vectors calculated by the SVM algorithm are written to the FPGA (7) in time for the start of the new switching cycle (8). Once these critical tasks have been completed, the remainder of

the data is sampled and validated (9). The data measured and calculated during the current processing cycle is output to the D/A converters (10) and the serial data output buffer (11). The command input buttons are checked (12) ready for the next cycle, and the ISR terminates (13). Any requests received from the PIC via the SPI port are processed (14) in the 21μs of time remaining between the end of the ISR and the start of the next processing cycle due to the next ISR.

7.3.2 Data Acquisition

Every processing cycle a large amount of data is acquired by the DSP and written into its data table. As shown in Figure 7.8, the data acquisition is performed in two stages. At the start of the ISR the stator currents are sampled, while the remainder of the data is sampled once the new set of voltage vectors has been written to the FPGA. As the raw conversion results from various signals are retrieved from the ADC, they are converted to their real values using scale and offset factors stored in the configuration table. The IGBT desaturation fault status, over-current status, and trip status registers are also copied from the FPGA into the data table. The rotor position and speed information is also read from the FPGA as it becomes available.

In order to protect the power electronic hardware from damage, the acquired data values are compared against preset limits in the configuration table and the fault status bits are checked. If any data value exceeds its warning limit, the DSP turns on a yellow warning LED. If any data value exceeds its configured trip limit or if a fault status bit is set, the DSP enters a tripped state. In this state the control algorithms and data acquisition is suspended. This allows the cause of the trip and the state of the system at that time to be examined before the fault is cleared and the trip condition is reset.

7.3.3 Outer Control Loops

Once the motor currents have been sampled, the torque control algorithms are run. There are a number of cascaded PI controllers between the torque reference supplied by the user and the torque reference to the direct torque controller. These PI controllers attempt to enforce particular limits by overriding the users torque reference. Their inclusion is necessary to minimise nuisance trips of the controller. For example, the torque reference

is reduced as the electric vehicle nears the maximum speed limit such that the vehicle is limited to the maximum speed. The alternative would be for the controller to trip when the speed limit is exceeded, which would be inconvenient.

There are a total of 6 stages in the outer control loop.

1. Speed Control – If speed control is enabled, a PI regulator is used to control the torque reference in order to achieve a desired speed irrespective of motor load.

2. Input Saturation – The torque demand and stator flux inputs are saturated at their configured minimum and maximum values.

3. Torque Rate Limiter – Very rapid changes in torque reference require large voltages to be applied to the motor. If the required voltage cannot be generated the controller resorts to choosing a voltage vector that will drive the output torque towards the reference value. Precise control of stator flux cannot be achieved during this time. The torque rate limiter can prevent this situation from occurring.

4. Over-voltage Control – During braking conditions (negative torque demand while moving forward), the controller converts mechanical energy into electrical energy that is returned to the battery bank, causing the battery voltage to rise. As the voltage nears the upper limit, the braking torque demand is decreased to reduce the current going into the batteries and therefore the voltage rise.

5. Under-voltage Control - As the voltage nears its lower limit the torque reference is reduced in an attempt to reduce the current drain on the battery bank and therefore allow the battery voltage to increase.

6. Speed Limiting - As the motor speed nears an upper limit, the torque reference is decreased to reduce the speed.

7.3.4 Direct Torque Control

The torque reference generated by the outer control loop is passed to the direct torque controller stage that calculates the voltage needed to produce the torque reference. The sequence of equations required to calculate the voltage were summarised in Section 3.2.6.

To achieve perfect direct torque control it is necessary to correctly estimate the torque produced and then apply a voltage to force the torque to the reference value. This needs to occur instantaneously, but in practise this is not possible due to calculation delays. Therefore the calculations are performed ahead in time to enable the results to be ready for the next switching cycle. The instantaneous torque is estimated early and the voltage is calculated to drive that estimate to zero. By the time the voltage is applied, the actual torque produced will have changed. The result is a steady-state error in the torque that is produced.

Section 7.3.1 described how the calculation delay was reduced from 100μs to 50μs by delaying the start of the processing cycle to the middle of the switching cycle. Using the model of the induction motor, the actual current measurement that was taken early can be projected forward by 50μs using equation (3-15) to estimate what the current will be at the end of the switching cycle. This projected current is used to predict the torque at the start of the next switching cycle and therefore the voltage required to correct the predicted torque error. This technique minimises the steady state torque error due to the calculation delay.

The voltage prediction described in Section 3.2.3 is based on the instantaneous state of the induction motor. In practice, the state of the induction motor will change over the next switching cycle while the voltage is being applied. Since the rotor is continuous rotating forward, so are the rotor flux and the back EMF. Over the period of the switching cycle, the back EMF will have rotated towards the applied voltage and the actual change in current will be less than that predicted by equation (3-15). Therefore, the change in torque will be less and a steady-state torque error will result. This effect becomes more significant at high speed were the back EMF is moving further within each switching period. The effect of the motor state changing over the switching period can be

observed in the simulation results of Figure 5.1. For a torque reference of 5Nm at 2000RPM, an average torque of 4.3Nm was produced.

7.3.5 Space Vector Modulator

The voltage calculated by the direct torque controller stage is synthesised using space vector modulation. Full details of the space vector modulation algorithm were provided in Section 4.2. Some of the details more specific to the actual implementation are described in this section.

Comparisons are performed to determine the sector and sub-sector that the reference voltage vector lies in. These are used with the DC bus equalisation state to determine the required vector sequence using a lookup table with 48 entries, two for each of the 24 sub-sectors. Each entry specifies the vectors and their sequence. Each of the 27 possible voltage vectors is represented by a unique id code. It is possible to operate the three-level inverter in a two-level mode using an alternate lookup table that ignores the sub-sector and DC bus equalisation indices and returns only those vectors that a two-level inverter would generate. These vectors are a sub-set of the voltage vectors that a three-level inverter can generate.

The vector sequence information and a lookup table are used to determine vectors of unit length that are then multiplied by the instantaneous DC bus voltage to obtain the actual voltage vectors that will be generated by the inverter. Equation (4-3) is then solved in real-time using a form of Cramer's Rule for solving systems of linear equations [Anton, 1994]. This requires one division operation, 18 additions/subtractions, 36 multiplications, and takes 3.0μs to execute on the DSP. Because of calculation errors, it is possible that very short duty cycles may actually be calculated as negative. To avoid problems later in the algorithm due to negative values, any duty cycle that is calculated as less than 10^{-6} is set to zero. A duty cycle of less than 10^{-6} is below the resolution of the FPGA timing generation logic.

The duty cycles are converted into the required on-duration count by multiplying by the switching period and the FPGA counter clock frequency. Any vector that is below the configured minimum on-duration is discarded and its time is split amongst the remaining

vector(s). All the remaining vectors are written to the FPGA in the required sequence. The first vector of the second sub-cycle is flagged such that it generates an interrupt to the DSP in order to run the next processing cycle at the appropriate time, as described in Section 7.3.1.

The FPGA handles the remainder of the process of generating the IGBT timing signals once the vector id codes and their on-duration counts have been loaded into it. The actual output voltage considering the dead-time effects is then estimated as described in Section 4.4.3 and is fed back to the torque control loop that then compensates for the dead-time.

7.3.6 Data Output

After all the control algorithms have been run and the FPGA has been loaded with the data for the next cycle, selected data from the DSP is output via the D/A converters and the serial port.

The configuration table contains information on the allocation of each DAC channel. This includes the index of the item in the data table to output and the minimum and maximum values. At the end of the processing cycle, the required data items are retrieved from the data table, scaled based on the configured minimum and maximum values and written to the FPGA which then updates all the DACs. During operation of the torque controller, the Motor Manager software can be used to change the data that is output and the scale by changing values in the configuration table.

Selected values from the data table can also be sent over an RS232 serial connection to a PC for analysis using the Motor Monitor software (described in Section 7.4). Due to the limited capacity of the serial connection, the DSP maintains lists of data values to be output each cycle. The appropriate list of data is selected depending on the requirements of testing. The DSP processes the list at the end of the processing cycle by taking each of the required values from the data table, scaling it to preserve the required number decimal points, and converting it to a 16-bit integer that is inserted into a buffer. There was sufficient memory available in the DSP for a buffer capable of storing 12 data values from consecutive cycles for a period of 76ms. Data collection stops when the output buffer becomes full, at which point all data is sent to the PC.

The format of the data packet sent over the RS232 connection is shown in Figure 7.9. This contains the data for an entire processing cycle represented as 16-bit integers. It also contains a frame id that the DSP uses to identify the data that the packet contains and a 32-bit sequence number that identifies the processing cycle that generated the data. It is also possible for the DSP to immediately send the data, although intermediate cycles are lost while the data is being transmitted.

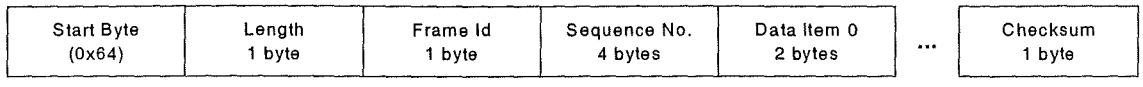


Figure 7.9: Serial Data Output Packet

7.3.7 Performance Issues

The DSP software was written entirely using the 'C' programming language with the exception of some simple macros for accessing the DSP status and interrupt registers that were written using in-line assembly code. No special optimisations, such as hand-coding critical pieces of code in assembly language, were necessary due to careful initial design of the software and the high performance of the TMS320VC33 DSP. The critical segment of code between sampling the currents and outputting the next IGBT switching sequence could be executed within 50μs, allowing the calculations for the next switching cycle to be delayed until the middle of the current cycle. This could be achieved even with all complier optimisations turned off, simplifying debugging of the code.

The floating-point unit of the C3x series DSPs does not support floating point divisions, which are instead performed using a library function involving numerous multiplications and addition/subtraction. The number of divisions was kept to a minimum by rearranging the code, or if division by a specific value was commonly used, calculating the inverse of that value once and then multiplying by the inverse whenever a division was required.

Extensive use of digital logic in the FPGA to implement some functions also reduces the processing load on the DSP. An example of this is the IGBT gate signal timing generation. Once the mathematical calculations have been performed, the rest of the process is off-loaded to the FPGA. The alternative would be to use an internal DSP timer

to interrupt the DSP every time the gate signals needed to change, which is an inefficient use of the DSP's capabilities.

7.3.8 Simulation of the DSP software

Once the initial development of the DSP software was complete, it was tested using MATLAB and Simulink against a simulated induction motor. This enabled the control algorithms to be verified before they were introduced into the more complex experimental system. The core parts of the DSP software were incorporated into Simulink by building them into a MATLAB S-function in the form of a Dynamically Linked Library (DLL). The core components simulated were the configuration and data tables, the outer control loops, and the direct torque control and space vector modulation algorithms. A simple interface for these components was written to enable the DSP code to be used as a MATLAB S-function.

When MATLAB initialises the S-function, it calculates an initial set of voltage vectors that are queued in the PCs memory rather than at the FPGA. At each time-step the S-functions 'Output' function is invoked and is passed the motor stator current, the torque and flux references, and the DC bus voltage from the simulation. The 'Output' function de-queues and returns a voltage vector that is applied to the simulated induction motor. It also calculates the next time-step as the current simulation time plus the on-duration of the current vector. If it is the appropriate simulation step, then the control algorithms are re-run to create the next set of voltage vectors.

7.4 Motor Monitor Application

The Motor Monitor software is a PC based application for the display of data that has been output by the DSP via its RS232 interface. Like the Motor Manager software, the Motor Monitor application was created using Visual C++ and MFC and has the following features.

- Numerical display of data values received
- Graphing of scalar data values

- A d,q plot of vector data values
- Recording of received data
- Playback and single-stepping through the recorded data

Figure 7.10 shows the main window of the Motor Monitor application. In the top left 'Raw Data' window is a numerical display of the data that was received from the DSP. It shows that five vector values (displayed as D and Q components) and two scalar values were received. In the lower left 'Scalar Plot' window is a graph of the two scalar values (Torque Out and Torque Delta) over the last 10ms. In the 'Vector Plot' window on the right is a d,q plot of the vector values. Both of the plot windows can be configured to determine which incoming data values are displayed, the number of grid lines, and the scale used on the axes.

Extra information about the vectors is shown on the left hand side of the vector plot window. This consists of the magnitude of the first three vectors selected and the phase angle in degrees to each of the next three selected vectors. For example, Figure 7.10 shows that the stator current vector (I) is 59.93A in magnitude, the voltage vector is 64^0 ahead, flux is 37^0 behind, and the EMF is 34^0 ahead.

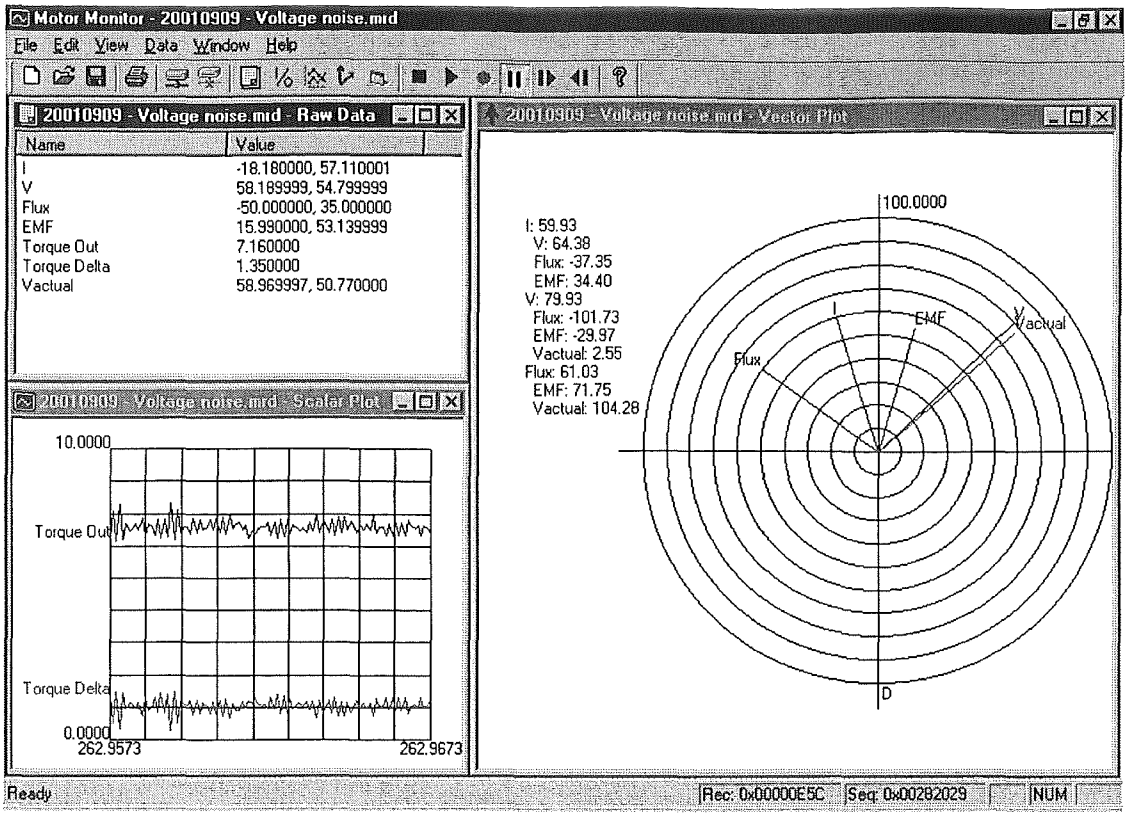


Figure 7.10: Motor Monitor Application Main Window

In the following sections, Section 7.4.1 describes how the incoming data is decoded and Section 7.4.2 describes how the Motor Monitor application is used to analyse the operation of the torque controller.

7.4.1 Data Reception

The DSP sends measured and calculated data over an RS232 serial connection using the data format described in Section 7.3.6. The DSP is able to output different sets of data with the Frame Id code being used to identify the format of the data that the packet contains. The Motor Monitor application is designed to support the addition of new frame ids to the DSP software by maintaining records that define how the incoming data packet is decoded into data values. The configuration of these records is achieved through a collection of dialog boxes.

Figure 7.11 shows the 'Edit Record' dialog box that is used to define the encoding of a incoming data packet with a frame id of '2'. Using this dialog box the byte offset of each data item from the start of the encoded data is specified.

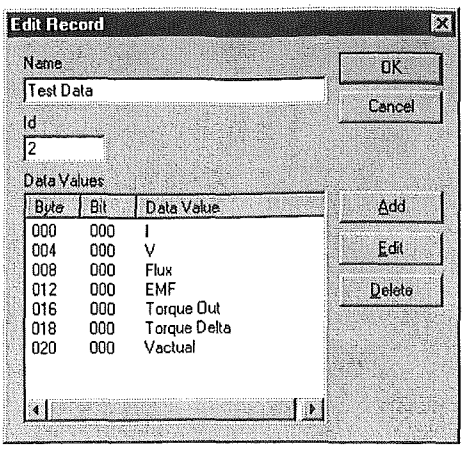


Figure 7.11: 'Edit Record' Dialog Box

Each data item is specified using the dialog box shown in Figure 7.12. This dialog box determines how the data is decoded from the incoming data packet, the scale factor to convert it back to its original floating-point value, and the colour to use when it is plotted. The data is normally scaled by the DSP to preserve the required number of decimal points and then converted into a 16-bit integer. When the PC receives the data it is scaled back to its original value. Vector values are encoded as two consecutive 16-bit integers.

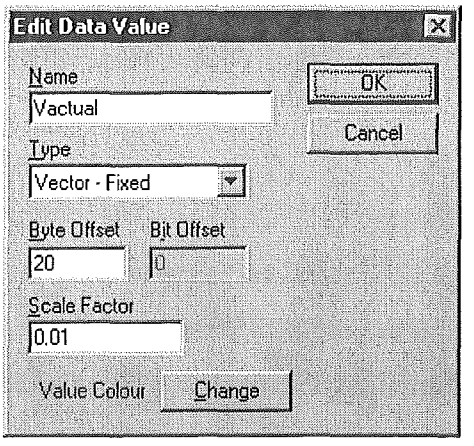


Figure 7.12: 'Edit Data Value' Dialog Box

When a valid data packet is received, the frame id is decoded and the associated record is retrieved. The Motor Monitor application then works through the list of data items in that record and decodes the value for each data item from the received data packet. The three different views are then notified that new data has been received so they can redraw themselves based upon the new data.

7.4.2 Data Analysis

As described in Section 7.3.6, the DSP is normally configured to buffer data from consecutive processing cycles and then send these to the Motor Monitor application in bursts. When required, the Motor Monitor application is used to record a collection of these data bursts. The recorded data can be played-back, saved for later analysis, or written into a '.MAT' that can be loaded into MATLAB.

To analyse the operation of the torque controller, the recorded data can be played-back in slow motion to observe the movement of the vectors. When an event of interest is observed, playback can be paused and the data values can be examined in detail in the Raw Data window. It is then possible to step forward and backward through each cycle to see how things change.

The ability to collect data, display as it vectors, and then to step from cycle to cycle was very helpful in the development and testing of the DSP software. The vector display is significantly better than that which can be achieved using the X-Y display mode of an oscilloscope.

7.5 Summary

This chapter has described the four components of software developed for this project. Two components run on the processor board. These are the System Control software on the PIC microcontroller that has overall control of the processor board, and the Motor Control software on the DSP that performs the torque control. The other two components of software run on the PC and communicate with the processor board using RS232 serial connections. The Motor Manager software communicates with the System Control software and is used to configure and control the system. The Motor Monitor application captures and displays information output by the Motor Control software running on the DSP.

The core control routines used in the DSP software were tested by building the 'C' code into a MATLAB S-function that was tested against a simulated induction motor. Once

the control algorithms were verified against a simulated motor, they were ready to be tested in a real system as described in the following chapter on the experimental results.

8. Experimental Results

The previous chapter described how the implemented control algorithms were simulated using MATLAB and Simulink. This chapter describes the experiments performed to validate the torque control and SVM algorithms, and the embedded controller and power electronic hardware against a real induction motor.

Section 8.1 explains the experimental set-up that was used for testing. Section 8.2 presents the results of open-loop testing of the space vector modulator and three-level inverter. These compare the current distortion for two and three-level operation. Section 8.3 shows the results of closed-loop testing of the torque controller. The steady-state performance and response to step changes in the reference signals are examined.

It was discovered during testing of the torque controller that the 400Hz induction motor used for testing had a defective rotor, resulting in very high slip speeds. The problems with the motor are described in Section 8.4. Because of these problems, the results presented were obtained while operating below 6000RPM, such that the excitation frequency was below 400Hz.

8.1 Experimental Set-up

The torque controller and three-level inverter were tested using a 200V, 400Hz, 15hp induction motor. A variable voltage DC supply is obtained using a three-phase variac and rectifier with a large bank of capacitors. Mechanically coupled to the test motor is a 5kW induction motor that is driven by a commercial motor drive and is used as a controllable mechanical brake. A photograph of the experimental set-up is shown in Figure 8.1 and a corresponding block diagram is shown in Figure 8.2.

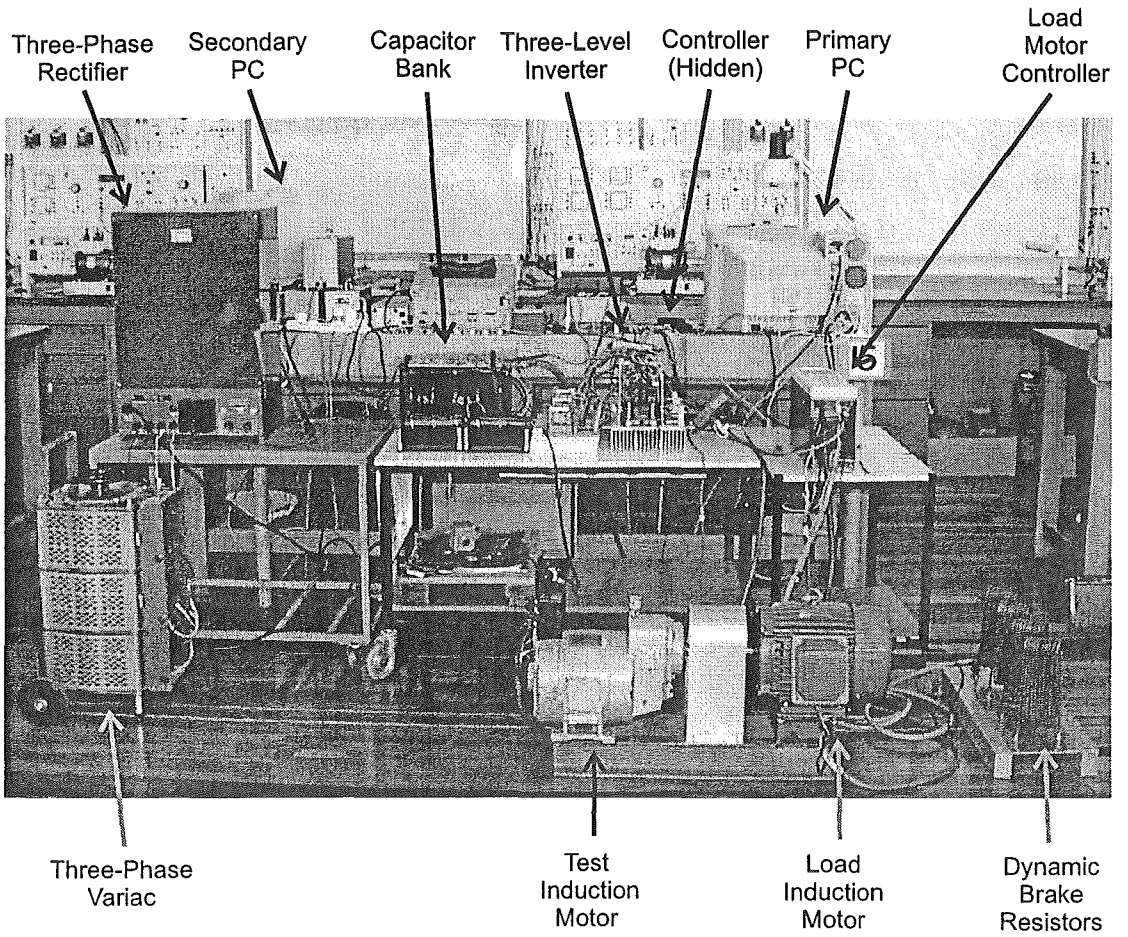


Figure 8.1: Photograph of the experimental set-up

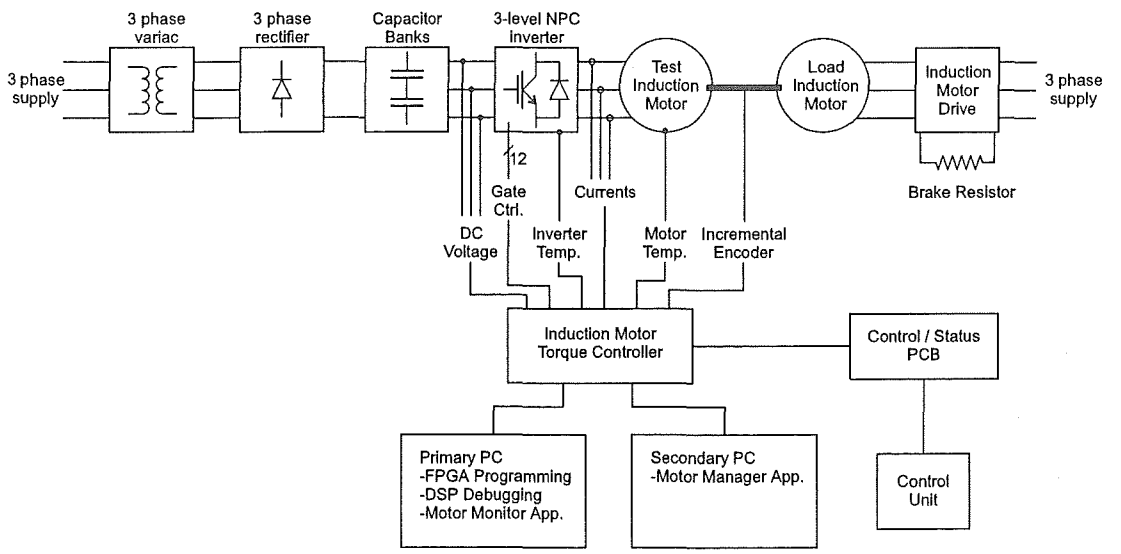


Figure 8.2: Diagram of the experimental set-up

A 25amp, three-phase variac is used to provide a variable three-phase voltage. The output of the variac is converted to DC using a three-phase rectifier. On the output of the

rectifier are ten 5100μF capacitors arranged as two series connected banks of five capacitors each. These capacitors provide the DC bus, with the mid-point between the two capacitors banks being the neutral point required by the three-level inverter.

A 600V, 150A (peak) three-level neutral point clamped IGBT inverter [Li, 1999] is used to drive the test induction motor. The inverter has twelve IGBTs, each with an associated gate drive circuit. Each gate drive circuit is connected to the torque controller and has its own floating power supply. The torque controller measures the motor stator current at the inverter output using three LEM LA-100S current transducers.

The induction motor used for testing is a 400Hz, 200V, 4 pole, 15hp (11kW) induction motor that has an integrated reduction gear with a ratio of 4.6667:1. At the rated speed of 11,500RPM, the output shaft speed is a more manageable 2,470RPM. The parameters of the test motor were obtained from no-load and blocked rotor tests. The test results and the parameters are listed in Appendix E. The test motor is connected to the inverter using screened cable to minimise interference caused by the high frequency switching of the inverter. The controller measures the motor frame and inverter heat-sink temperatures using LM35 temperature sensor ICs to protect against overheating.

The torque controller generates the switching patterns the three-level inverter. It measures the inverter DC voltage, the motor current, and the motor speed and uses this information to generate the inverter gate switching signals. The motor speed is measured using a 1,000 pulse per revolution incremental encoder on the shaft of the load motor. This encoder is used to display the motor speed, for over-speed protection, and by the current model stator flux observer. No encoder is required if the voltage model stator flux observer is used. Connected to the controller are two PCs. One PC runs the FPGA programming software, the debugger for the DSP, and the Motor Monitor application that displays and captures data from the DSP. The other PC runs the Motor Manager application that enables the performance to be monitored and the configuration to be changed in real-time. Also connected to the controller is a control and status PCB. This has the three status LEDs, start / stop buttons, and a torque reference control knob.

A standard 50Hz, 5kW induction motor is mechanically coupled to the test motor and is used as a controllable mechanical load. This load motor is driven by a Microdrive Elite ME-22.5 induction motor drive from PDL Electronics. The load motor drive is fitted with an external dynamic brake resistor that has a 10kW continuous rating. The motor drive is normally operated in speed control mode with an adjustable speed set-point. The motor drive is configured with a maximum torque limit of zero, such that it cannot drive the test motor and can only act as a brake by producing negative torque to keep the load motor at the speed set-point. The torque output of the test motor is calculated as the braking torque that is required plus the frictional and windage torque at that speed.

8.2 Space Vector Modulator Performance

This section examines the performance of the three-level space vector modulator. It compares the current harmonics produced for two-level and three-level operation. As discussed in Section 7.3.5, two-level operation of a three-level inverter can be achieved by ignoring voltage vectors that use the zero voltage state. Section 8.2.1 compares the current distortion as a function of voltage amplitude and frequency. Section 8.2.2 compares the current frequency spectrum over the first 20 harmonics and Section 8.2.3 compares the current spectrums at higher frequencies, including the switching frequency.

8.2.1 Comparison of Two and Three-Level Operation

Three-level inverters have a number of advantages as discussed in Section 2.1.1. The main advantage in this application is the reduced distortion in the synthesised voltage, which results in lower current harmonics. These current harmonics represent wasted energy since they don't generally contribute to torque production. This section compares the current distortion as a function of voltage amplitude and frequency for two and three-level operation.

For an induction motor operating under steady-state conditions, the voltage amplitude / frequency ratio is normally constant in order to maintain a constant level of flux. To obtain a realistic evaluation of space vector modulation and the inverter, the current distortion as a function of voltage and frequency is examined where the voltage amplitude to frequency ratio was constant.

The distortion of the synthesised voltage normally decreases as the voltage amplitude increases and the frequency decreases. As the fundamental voltage amplitude increases, the relative amplitudes of the harmonics compared to the fundamental decrease. Also, the desired voltage waveform is synthesised from a number of discrete steps, with each step corresponding to a single switching cycle. As the frequency of the voltage increases, there are fewer switching cycles per fundamental period and therefore each step is larger, resulting in a more distorted voltage waveform.

Figure 8.3 shows a plot of total harmonic distortion (THD) as a percentage of the fundamental current as the modulation level is increased from 0 to 1. Here the modulation level is defined as the ratio of voltage amplitude to frequency where a modulation level of one corresponds to the maximum frequency and voltage that the inverter can produce. In this case, the maximum frequency was 400Hz and the voltage was 170V rms line-to-line. This differs from the modulation index, which is defined as the voltage amplitude relative to the maximum voltage amplitude for a fixed frequency. The THD measurement was taken over the first twenty harmonics of the fundamental component of current.

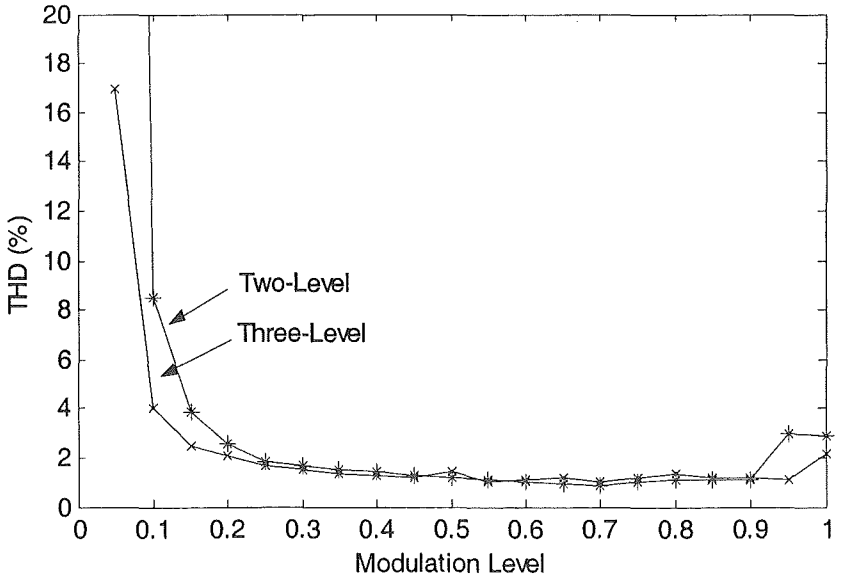


Figure 8.3: Current distortion as a function of modulation level

For a modulation level of 0.05 (corresponding to 20Hz, 8.5V rms line-to-line), the THD for two and three-level operation were 161% and 17% respectively. The three-level inverter was significantly better but 17% THD is still very high. At 0.1, Figure 8.3 shows the THD for two-level operation is twice as high, and at 0.15 is 50% higher. Above 0.25, the current distortion is very low for both two and three level operation and the difference between the two is negligible. At high modulation levels (0.95 and 1.0), the current distortion begins to increase again.

This increase in current distortion is due to the minimum on-time limitation coming into effect. The maximum sinusoidal voltage that can be produced using space vector modulation corresponds to a d,q voltage vector trajectory that just fits inside the hexagon of Figure 2.3. The voltage limit is at the mid-point of each sector. As the voltage nears the limit, the on-duration of the zero vector decreases below the minimum and is discarded, resulting in a more distorted waveform. This effect is not as significant for a three-level inverter since the six inner voltage voltages are used when generating large voltages, rather than the zero voltage vector. The three-level inverter can generate higher voltages before it needs to start discarding vectors due to the minimum on-time limitation.

8.2.2 Current Harmonics

As shown in the preceding section, there is only a measurable difference in the current distortion for modulation levels below 0.25 and above 0.9. This section examines the magnitudes of the current harmonics at a modulation level of 0.1, which corresponds to a frequency of 40Hz. Figure 8.4 and Figure 8.5 show the frequency spectrum of the phase current from 0 to 800Hz for two and three-level operation respectively. These frequency spectrums cover the first 20 harmonics of the 40Hz fundamental. Each harmonic is marked with a 'v' symbol. As expected the 5^{th}, 7^{th}, 11^{th}, 13^{th}, 17^{th} and 19^{th} harmonics are presented and decrease in amplitude as the harmonic number increases. Table 8.1 shows a comparison between the amplitudes of these harmonics for two and three-level operation. Three level operation resulted in an approximately 10dB reduction in the 5^{th} and 7^{th} harmonics. The amplitudes of the 13^{th} and 17^{th} harmonics increased slightly, but are still 40dB and 50dB respectively below the fundamental component.

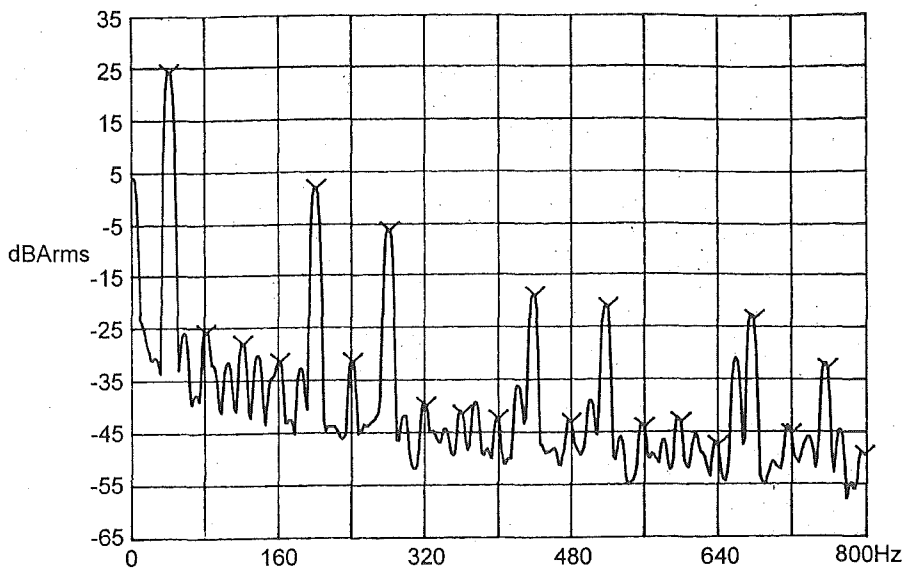


Figure 8.4: First 20 current harmonics for two level operation

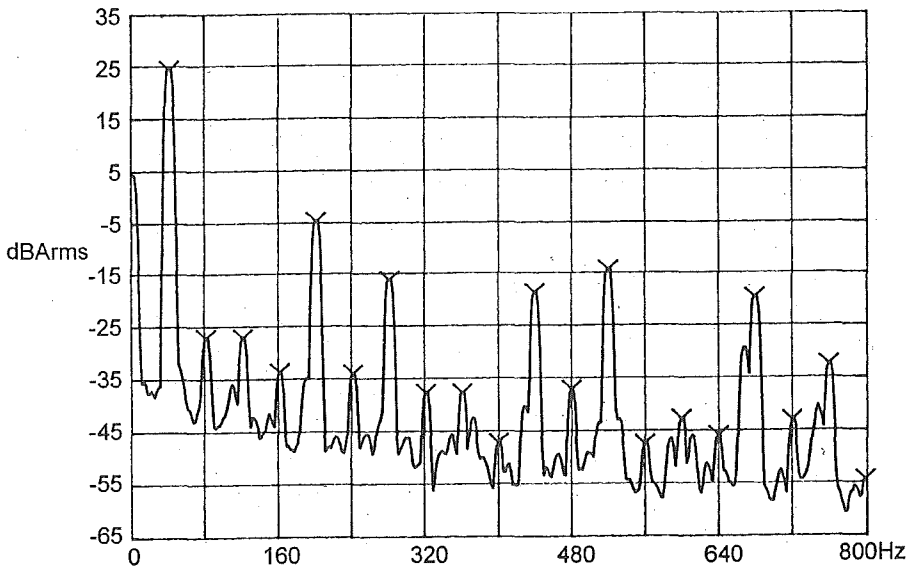


Figure 8.5: First 20 current harmonics for three level operation

Table 8.1: Comparison of harmonic amplitudes

Harmonic Number	Two-Level Operation	Three-Level Operation	Decrease in Amplitude
1	26 dB	26 dB	0
5	3 dB	-5 dB	-8 dB
7	-6 dB	-16 dB	-10 dB
11	-19 dB	-19 dB	0
13	-21 dB	-14 dB	7 dB
17	-23 dB	-20 dB	3 dB
19	-32 dB	-32 dB	0

8.2.3 Switching Harmonics

Figure 8.6 and Figure 8.7 show the spectrums of the current harmonics from 0 to 50kHz for two and three-level operation respectively. The fundamental frequency was reduced to 10Hz so that the fundamental frequency and its associated harmonics are lower than the spectrum analyser bandwidth of 477Hz and therefore appear as a single peak. Figure 8.6 for two-level operation shows only harmonics at multiples of the 10kHz switching frequency. Figure 8.7 for three-level operation shows harmonics at multiples of 5kHz.

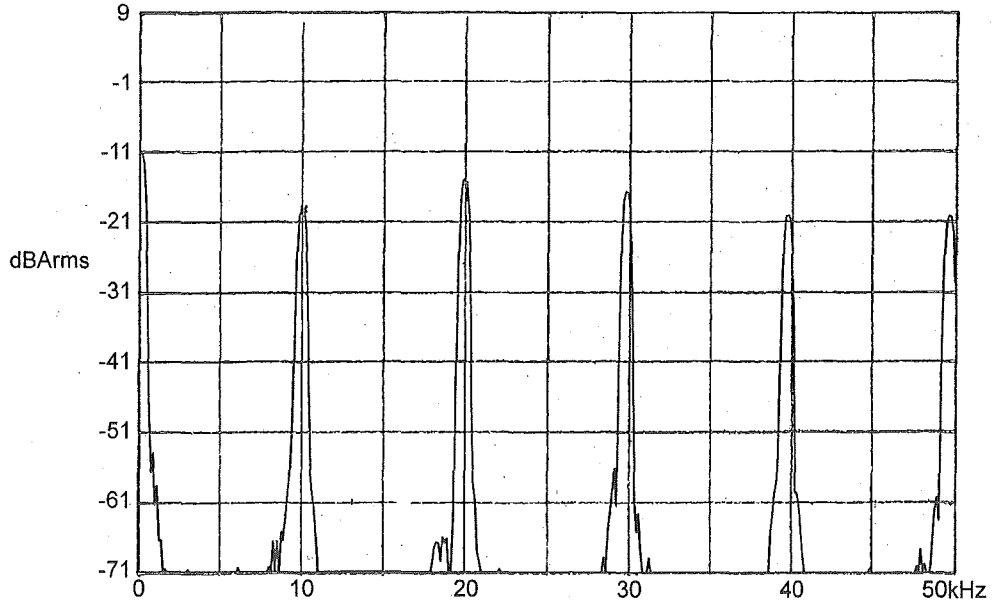


Figure 8.6: Switching harmonics for two level operation

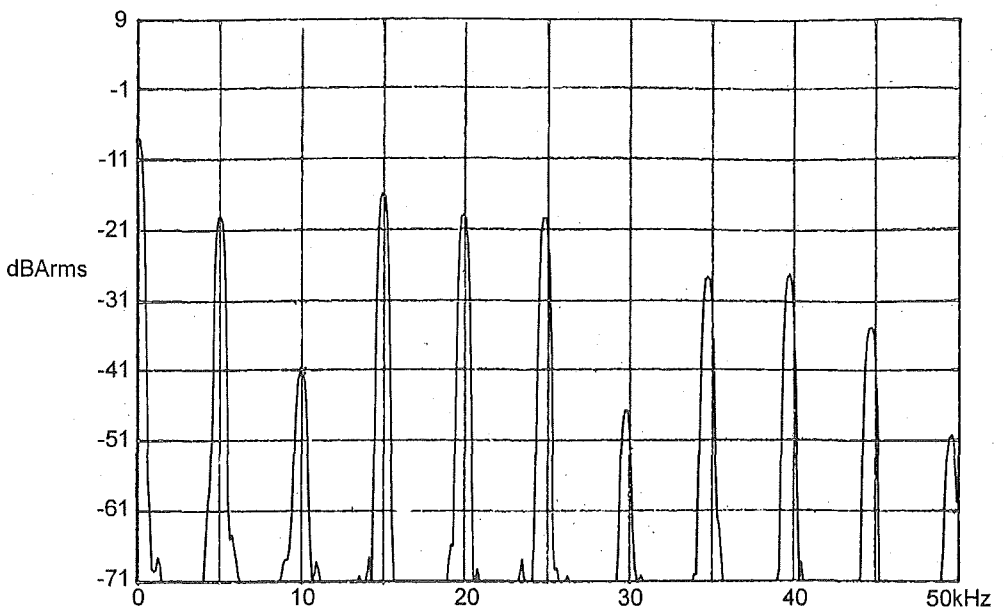


Figure 8.7: Switching harmonics for three level operation

To keep the two halves of the DC bus balanced, the space vector modulator will alternate each switching cycle between two sets of inverter states that produce the same line-to-line voltages but have different effects on the neutral point voltage. Section 4.2.2 explained how the order of the vectors is chosen such that only one phase needs to switch to change to a different vector. Due to this requirement, the two sequences of line-to-line voltages produced are different.

For example, to produce a voltage vector that resides in sub-sector 1 of sector 1, the following two switching sequences are used:

Sequence 1: `000, +00, ++0, +++, ++0, +00, 000`

Sequence 2: `000, 00-, 0--, ---, 0--, 00-, 000`

The inverter states `+00` and `0--` produce the same line-to-line voltage and likewise so do `++0` and `00-`. There are only three different line-to-line voltages produced, which can be defined as:

Voltage 0: `---`, `000`, and `+++`

Voltage 1: `+00` and `0--`

Voltage 2: `++0` and `00-`

The two switching sequences can be written as the following sequence of voltages:
Sequence 1: 0, 1, 2, 0, 2, 1, 0
Sequence 2: 0, 2, 1, 0, 1, 2, 0

To keep the DC bus balanced, the space vector modulator will alternate between these two sequences. Therefore the following sequence of voltages is used over a period of two switching cycles:
0, 1, 2, 0, 2, 1, 0, 2, 1, 0, 1, 2, 0

This is a non-periodic sequence of voltage that occurs over a period of two switching cycles (200μs), i.e. it occurs at half of the switching frequency.

If the switching sequences are temporarily changed such that one of the sequences ignores the requirement that only one phase must switch per vector transition, then both switching sequences can be arranged to produce the same sequences of voltages. The resulting frequency spectrum is shown in Figure 8.8. Compared to Figure 8.7, the peaks at 5, 15, 25, 35, and 45kHz have decreased significantly. They have not been removed completely since asymmetries since exist between the two switching cycles, such as the effect of the dead-band and the slight voltage unbalance between the two halves of the DC bus. The cause of these harmonics is also validated by the simulations results that were shown in Figure 5.4. The current frequency spectrum for field oriented control does not show an harmonics around 5kHz, while the more detailed model used for the DTC using space vector modulation simulations, which includes the DC bus equalisation effect, shows switching harmonics around 5kHz.

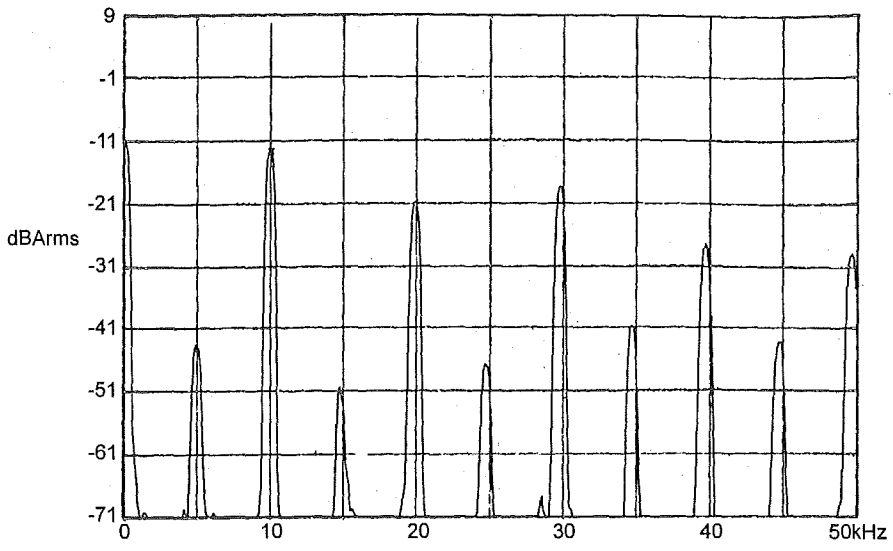


Figure 8.8: Switching harmonics for a modified switching sequence

Figure 8.8 showed the amplitudes of the harmonics at multiples of 5kHz decreased significantly when one of the switching sequences was modified. This is not a permanent solution as changing one of the switching sequences has increased the switching frequency as more than one phase must now switch at each transition to maintain the same sequence of line-to-line voltages.

8.3 Torque Control Performance

This section examines the performance of the direct torque controller when operating on a 400Hz induction motor. Sections 8.3.1 and 8.3.2 examine the steady state voltage and the torque error, while Sections 8.3.3 and 8.3.4 examine the response to step changes in the torque and stator flux reference values.

8.3.1 Steady State Voltage Reference

The direct torque control technique calculates the required voltage, which is then synthesised using space vector modulation. In order to produce constant torque and a constant flux magnitude during steady state operation, the stator flux vector most follow a circular trajectory at a constant speed. This is achieved using a voltage vector that also follows a circular trajectory at a constant speed. Any deviation from this circular trajectory is due to errors in the torque controller.

Figure 8.9 shows a plot of the calculated voltage in a d,q reference frame. The controller was set at 3Nm of torque and 0.047wb of flux with the motor running at 6,000RPM. The trajectory is approximately circular. In practise the voltage reference deviates slightly in order to compensate for the voltage error introduced by the inverter dead-time.

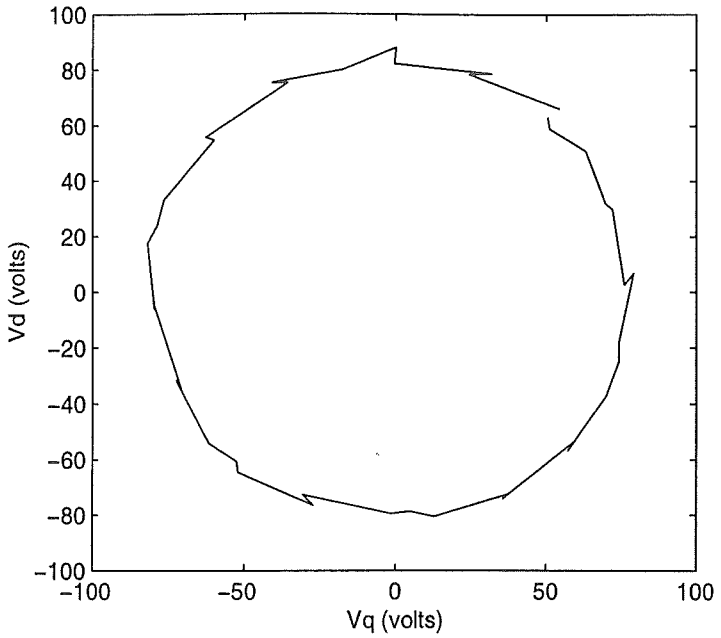


Figure 8.9: Voltage vector trajectory during steady state operation

8.3.2 Steady State Torque Error

Direct torque control calculates the voltage required to drive the estimated torque and stator flux to the reference values. The accuracy of the torque produced is dependant on the accuracy of the torque estimate, since it is the error in the torque estimated that is driven to zero. Equation (2-17) showed that the torque is estimated using only the stator flux and stator current. Therefore the accuracy of the torque produced is dependant on the accuracy of the stator flux estimate.

The steady state performance of the direct torque controller was determined by comparing the differences between the reference torque, estimated torque, and the actual braking torque required. The braking torque required will also contain errors since it is only an estimate taken from the load controller. The torque and flux references were set to 6Nm and 0.06wb and the motor speed was varied from standstill to 5,600RPM by controlling

the braking torque applied by the load motor. The test was repeated using both the voltage and current models for the stator flux observer with the results shown in Figure 8.10 and Figure 8.11 respectively.

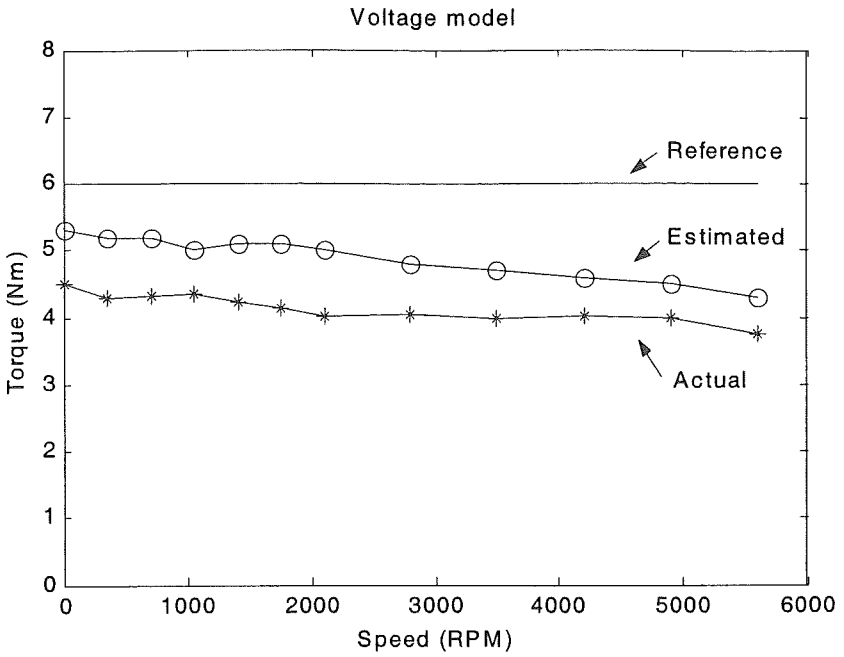


Figure 8.10: Torque error using the voltage model stator flux observer

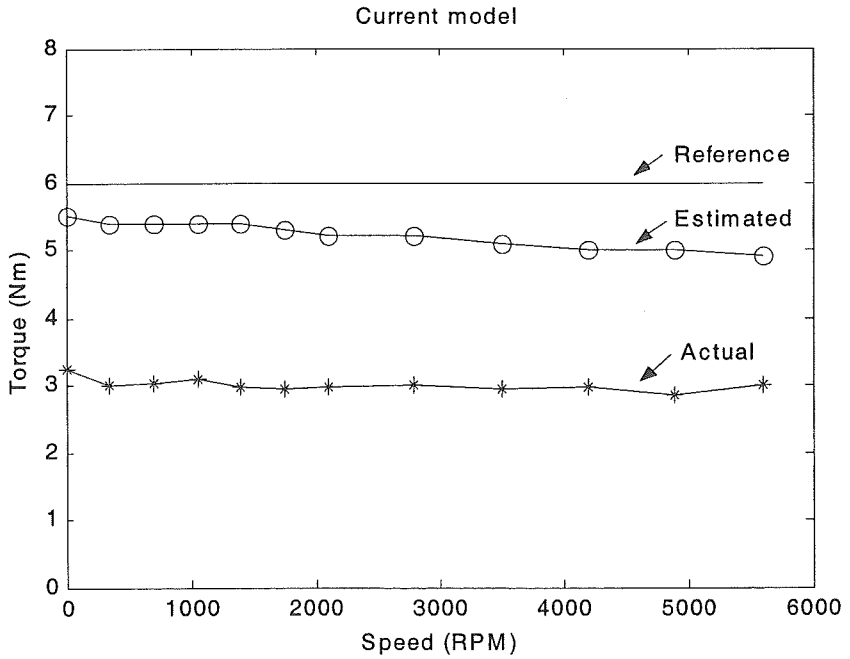


Figure 8.11: Torque error using the current model stator flux observer

These figures show there is a difference between the reference torque and the estimate of the torque produced. This means that the applied voltage was not sufficient to drive the estimated torque to the reference value. As described in Section 7.3.4, the voltage is calculated based on the state of the motor at the start of the switching cycle. The state of the motor will change while the voltage is being applied. The change in the motor state is typically such that the voltage will produce less torque than expected. The faster the motor speed, the greater the change in the motor state over the switching cycle and the less torque actually produced. This effect can also be observed in the torque ripple simulation results shown in Figure 5.1. For a torque demand of 5Nm at 2000RPM, the average torque produced by DTC using space vector modulation was 0.7Nm lower. Figure 8.10 shows a torque demand of 6Nm produced a torque estimate at 2000RPM that was 1Nm lower.

These figures also show the actual torque produced is less than the estimated torque. Due to the motor's high slip characteristic (as described in Section 8.4), the excitation frequency to produce 6Nm of torque is relatively high, even at standstill. At a high excitation frequency the voltage model can estimate the stator flux relatively accurately. This explains the relatively small torque error present in Figure 8.10. The torque error in Figure 8.11 is larger due to differences between the induction motor parameters used and the actual motor parameters. As described in Section 3.2.1.2, the effect of these parameter errors becomes larger when operating near rated slip.

Figure 8.10 and Figure 8.11 show the estimated torque decreases slightly as the motor speed increases. For both stator flux observers, the error between the estimated torque and actual torque was relatively constant over the range of speeds tested. This error ranged from 11% to 19% for the voltage model and from 39% to 45% for the current model.

8.3.3 Torque Response

The torque response of the controller was examined by applying a step change in the torque reference. For these tests, the motor speed was fixed at 2,100 RPM by the load motor controller and the torque reference was stepped from 1Nm to 6Nm. The stator flux was set at 0.06wb and a rate limit of 10,000 Nm/sec was imposed on the torque reference.

As described in Section 3.2.5, the torque controller can maintain precise control over torque if there is sufficient voltage to drive the torque to the reference value over a single switching cycle. A torque change rate of 10,000Nm/sec or 1Nm per switching cycle was chosen as a rate that could be achieved under most operating conditions.

Figure 8.12 shows a plot of the torque reference, estimated torque, and measured phase A current. The change in estimated torque occurs almost instantaneously and there is an almost instantaneous change in the frequency, amplitude, and phase of the stator current during this time. This figure also shows the sinusoidal phase current during steady state operation on either side of the step change in torque. An enlarged view of the region where the change occurs is shown in Figure 8.13. This shows the torque reference rate limiter has ramped up the torque reference by 5Nm in 500μs, which is five consecutive switching cycles. The estimated torque precisely follows the increasing torque reference.

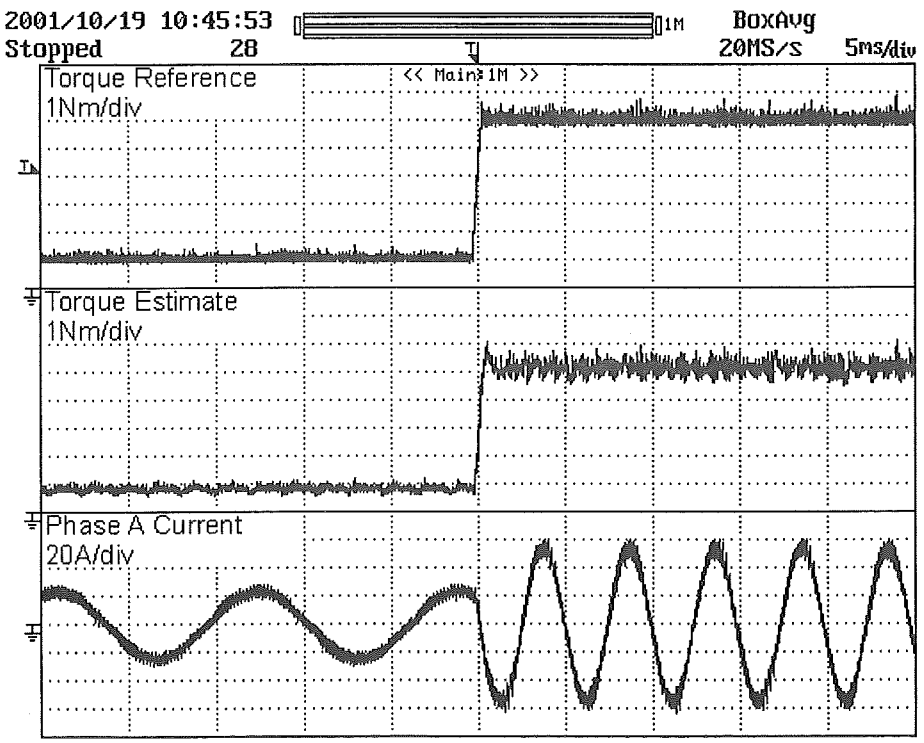


Figure 8.12: Torque Response - Rated limited to 10,000Nm/sec

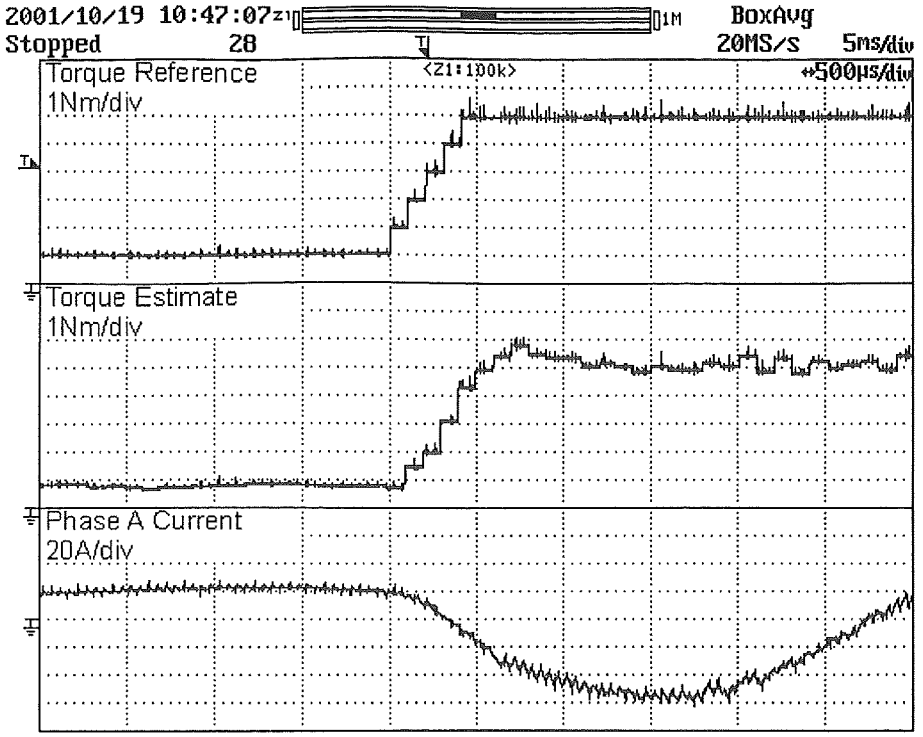


Figure 8.13: Torque Response - Detailed view of Figure 8.12

Figure 8.14 shows the same 1Nm to 6Nm torque step but with the torque reference rate limiting disabled. Again, the estimated torque responds almost instantaneously. Figure 8.15 shows an enlarged view of the region where the change occurs. The torque estimate increases to the new reference value over a period of 200µs, which is two switching cycles. Since the torque cannot be driven to the reference value in a single cycle, a single voltage vector is selected as described in Section 3.2.5 that will drive the torque towards the reference value. Figure 8.15 shows a rate of change of 30,000Nm/sec can be achieved under these operating conditions.

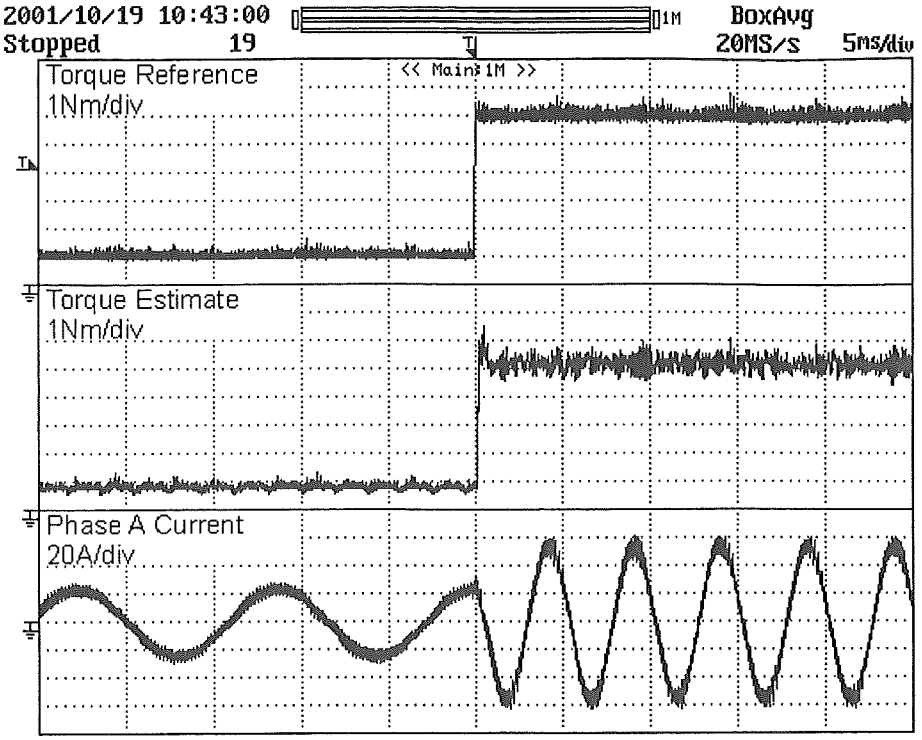


Figure 8.14: Torque Response - No rate limit

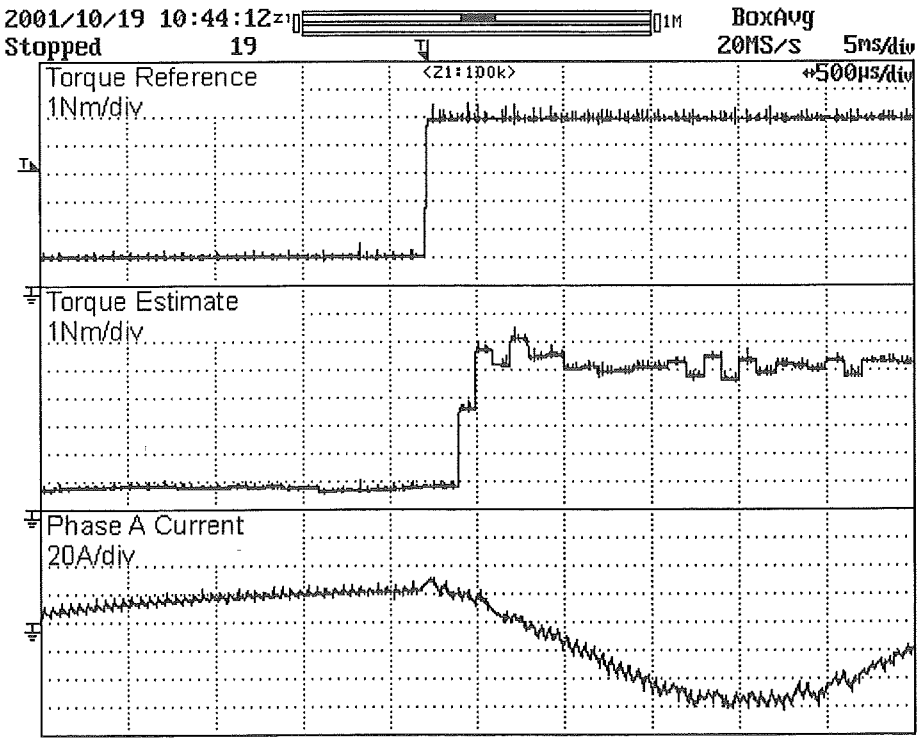


Figure 8.15: Torque Response - Detailed view of Figure 8.14

8.3.4 Stator Flux Response

The flux response of the controller was examined by applying a step change in the stator flux reference. The motor speed was fixed at 2,100RPM with a torque reference of 3Nm. The stator flux reference was stepped from 0.03wb to 0.05wb. Figure 8.16 shows the flux reference, estimated flux, current magnitude, and estimated torque. The estimated flux changes almost instantaneously since a single voltage vector is used to drive the stator flux towards the new reference value. A side effect of using a single voltage vector is the momentary loss of precise torque control as shown by the 3Nm increase in torque for a single cycle. The same 3Nm torque pulse can also be seen in the simulation results shown in Figure 5.8.

The plot of the current magnitude in Figure 8.16 is the magnitude of the stator current vector. This is calculated every 100μs switching cycle based on the currents measured by the LEM sensors. The plot shows a burst of current is required to increase the magnetic fields within the motor. The current decays as the flux of the rotor increases. Figure 8.16 also shows that less current is required when operating at the higher level of stator flux. The increase in current to maintain the higher level of flux is more than offset by the decrease in the torque-producing component of the current.

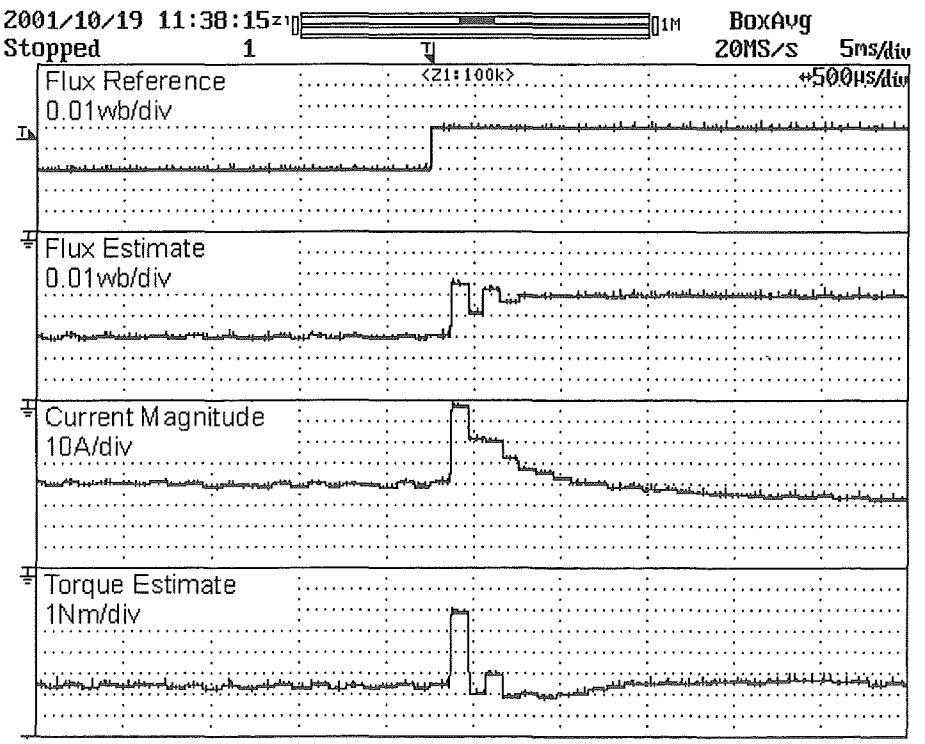


Figure 8.16: Stator flux response

8.4 Defective Induction Motor

While testing the direct torque controller as described in this chapter, some problems were discovered that were eventually traced back to the test induction motor. As the torque reference was increased, a point was reached were the calculated voltage exceeded the maximum inverter voltage and therefore the controller would revert to selecting a single voltage vector. At this point, the excitation frequency was over 400Hz even though the motor was only running at about half of its rated speed. To apply a voltage at a frequency greater than 400Hz and maintain rated flux, more than 200V is required. This indicated that something was wrong in the system if such a high frequency voltage was being applied while operating within rated conditions.

A simple open-loop test of the controller was performed. The controller was configured to produce a fixed sinusoidal voltage of 200V rms at 400Hz. The voltage and frequency were slowly ramped up over a period of 20 seconds to avoid high start-up currents. Once running, the load motor was controlled to supply a braking torque to decrease the speed. The speed was decreased from close to 12,000RPM, down to 7,350RPM while the torque output, power output, and current input were measured. The resulting torque/speed plot is called the slip characteristic and is shown in Figure 8.17.

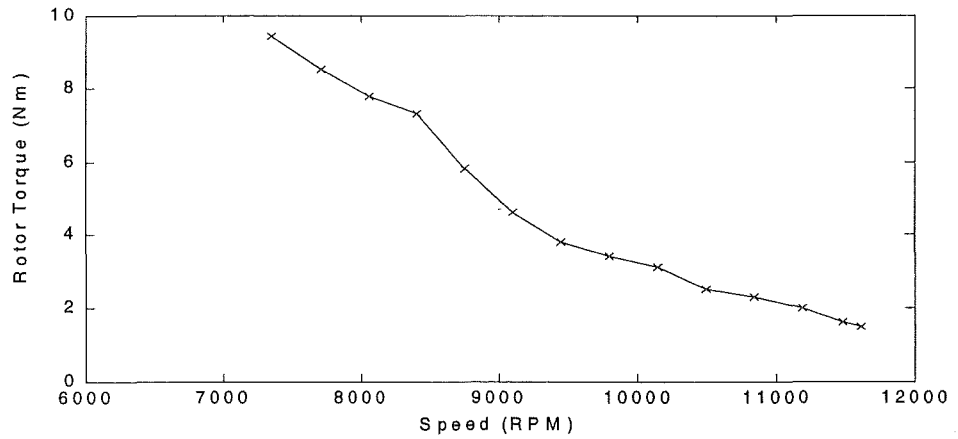


Figure 8.17: Slip characteristic of the test induction motor

At no-load, the motor ran close to its synchronous speed of 12,000RPM while drawing about 28A rms of current. When the motor was loaded with the rated torque of 9Nm, the

motor was running at 7,350RPM (39% slip) and drawing 62A of current. Because of the high slip (slow rotor speed), only 7kW of mechanical power was produced compared to the rating of 11kW. Such high slip speeds are not normal for an induction motor and can be a sign of high rotor resistance or under-fluxing due to insufficient voltage.

Figure 8.18 shows the stator phase current and its frequency spectrum from 0 to 1kHz while applying a voltage of 100V at 200Hz. The motor was loaded such that it ran at 4,900 RPM (18% slip). This corresponds to a rotor electrical frequency of 163Hz, or a slip frequency of 37Hz. The frequency spectrum in Figure 8.18 shows two very clear peaks near the fundamental current. The effect of the peak at 128Hz can be clearly seen as a low frequency modulation of the stator current waveform. Due to an electrical imbalance in the rotor, the induced rotor current has a negative sequence component. The magnetic field due to this negative sequence rotor current induces a current in the stator that is 37Hz below the rotor frequency [Bellini, Filippetti et al., 2001]. This current corresponds to the peak at 128Hz. The same effect would then repeat with this induced stator current, possibly causing the peak at 270Hz.

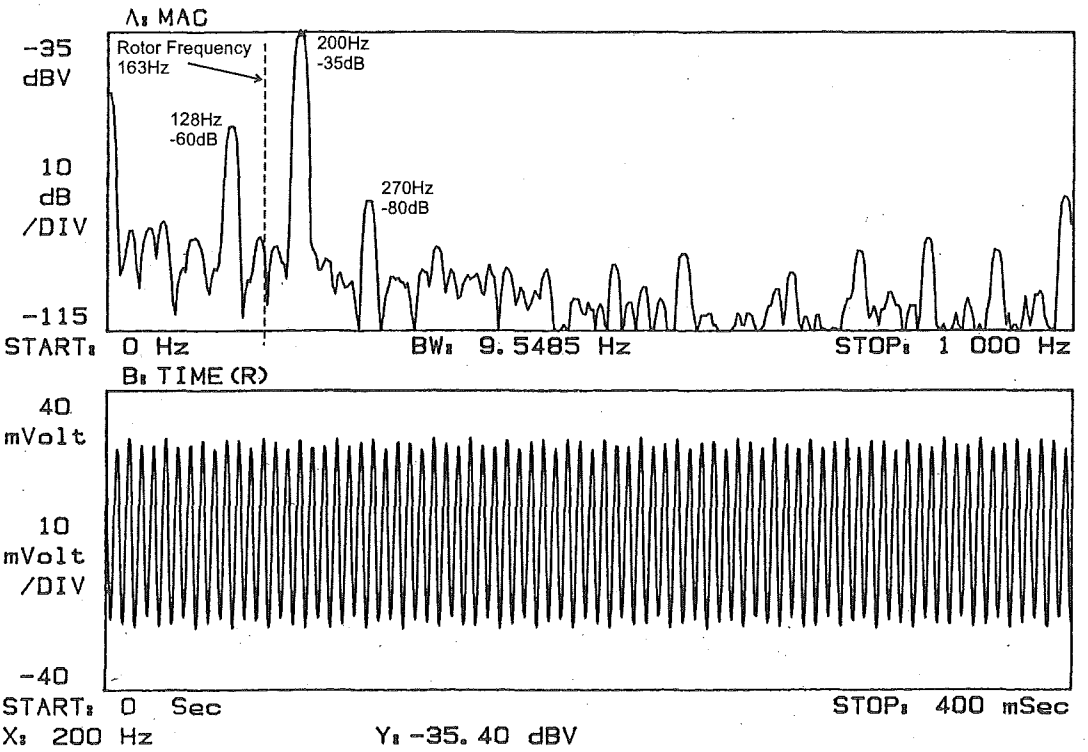


Figure 8.18: Test induction motor stator current at 100V, 200Hz

The test motor was disassembled to check for faults. The only possible problem noticed were some very shallow gouge marks in the centre of the rotor around half the circumference. These marks may have been caused by the motor overheating, which could cause the rotor to expand and come into physical contact with the stator. It is possible that this also damaged some of the rotor bars. This would increase the rotor resistance, explaining the high slip observed in Figure 8.17, and cause an electrical imbalance in the rotor that would induce a stator current as shown in the frequency spectrum of Figure 8.18.

The high slip prevented testing of the torque controller at higher levels of torque and speed due to the high excitation frequency and therefore the high voltage that is required. It may have also affected the accuracy of the motor parameters obtained, leading to a degraded stator flux estimate and the steady-state torque error that was observed. The induced stator current component due to the rotor electrical unbalanced did not appear when the torque controller was operating. The induced current component would have affected the stator flux estimated using the current model stator flux observer. The torque controller then applied a voltage to drive the stator flux in a perfect circular trajectory, effectively regulating out the induced current. Despite the problems with the test induction motor, it performed sufficiently to enable the direct torque controller to be evaluated with a high-speed motor.

8.5 Summary

This chapter has presented the experimental results that evaluated the performance of the three-level inverter and the direct torque controller. Compared to a two-level inverter, the three-level inverter showed a reduction in the current distortion when producing voltages with a modulation level below 0.25. When operating with modulation levels greater than 0.25, the current distortion was very low for both two and three-level operation. The three-level inverter caused switching harmonics to appear at 5kHz (half the switching frequency). These were not expected but occur due to the two different inverter switching-sequences needed to keep the DC bus balanced.

The direct torque controller had a steady-state torque error caused by errors in the estimated stator flux. Under the conditions tested, the voltage model stator flux observer caused less steady state torque error than the current model. The response to changes in the torque and stator flux references was almost instantaneous. A side effect of selecting a single voltage vector during a step change in the flux reference is the loss of precise torque control. The experimental results obtained under transient conditions match the simulation results shown in Figure 5.8.

During testing of the torque controller, it was discovered that the test induction motor had a defective rotor. A high slip characteristic was measured that is most likely due to a high rotor resistance caused by damaged rotor bars. An electrical imbalance caused by damaged rotor bars would also explain the currents induced in the stator windings.

The defective induction motor should not have greatly affected the results obtained, but it did prevent testing at high speeds due to the high excitation frequency required because of the high slip. The results that were obtained show that direct torque control using space vector modulation has low current distortion and a fast response to changes in the torque and stator flux references. A steady state error in the torque existed due to errors in the stator flux estimate.

9. Future Work

Some limitations of the torque controller were observed while it was being experimentally evaluated in the preceding chapter. These limitations include a steady-state torque error and loss of precise control over torque and flux during transient conditions. This chapter suggests future work that could be performed to improve the performance of the torque controller.

Sections 9.1 and 9.2 describes how improving the stator flux estimation and taking into account the induction motor core losses will reduce the steady-state torque error. Section 9.3 discusses changes to improve control during transient conditions and Section 9.4 describes how controlling the stator flux magnitude can improve the efficiency of the system.

9.1 Improved Stator Flux Estimation

Stator flux estimation is the currently the main limiting factor in the performance of the torque controller that has been presented. The estimated stator flux is used to estimate the torque that the motor is producing. Since it is the error between the torque estimate and torque reference that is controlled to zero, if the torque estimate is incorrect then the actual torque produced will also be incorrect. This can be observed in Figure 8.10 as a steady-state torque error. It will also affect the transient performance, but this could not be observed since a high bandwidth measurement of the actual torque was not possible.

Section 8.3.2 showed that the steady-state torque error is different depending on whether the voltage model or the current model is used for the stator flux observer. This difference means that the stator flux error was different for both observers. A self-tuning technique was described in Section 3.2.1.3 where the current model tunes the voltage model when running at low speed and the voltage model tunes the current model when running at high speed. Implementing this self-tuning technique would improve the stator flux estimation of both models and therefore reduce the torque error. Alternatively, a completely new stator flux estimator could be developed. This would ideally not require

the high-resolution encoder that is presently used to measure the rotor position for the current model.

9.2 Effect of core losses

As discussed in the introduction to this thesis, electric vehicle motors have a low inductance such that they run at high speed and therefore have a high power output relative to their size and weight. Due to their low inductance, the current ripple due to the inverter switching can be high, causing increased eddy current and hysteresis losses in the iron core. These core losses were not considered in the induction motor model that was used. Core losses will tend to degrade the steady-state torque accuracy and the dynamic performance [Matsuse, Taniguchi et al., 2001]. Including the effect of core losses in the induction motor model and modifying the controller to use this new model may improve its torque accuracy.

9.3 Improved Handling of Transient Conditions

The direct torque controller implemented used a simplified technique for handling transient conditions. If there was insufficient voltage available to drive the stator flux and torque to the new reference values in a single switching cycle, then a single voltage vector was selected that would drive both the torque and flux in the required direction. This technique can result in loss of precise control over the torque and flux for the duration of the transient. Section 3.2.5 described two techniques for handling the transient conditions. These were a calculation intensive approach that maintains full control over of torque and flux during transient conditions, and a simplified technique that provides good control over torque but sacrifices flux control performance. Using one of these techniques would enable improved control over torque and flux during transient conditions.

9.4 Flux Magnitude Optimisation

The flux level of an induction motor is normally maintained at the rated value up to rated speed and is then decreased linearly with increasing speed. This allows the maximum

motor torque capability to be achieved. It is possible to increase the efficiency of the system by controlling the flux level as a function of speed and torque. If the flux is reduced, the flux producing component of the current decreases, but the torque producing component increases as more current is required to produce the same torque at the lower level of flux. There is a flux level that achieves an optimal balance the flux and torque producing components of the current such that the total current magnitude and losses are minimised [Adnanes, Nilsen et al., 1993]. The flux optimisation calculation depends on the motor parameters, speed, and load. Due to the high dynamic performance of the direct torque controller, the flux can be rapidly restored to the rated value whenever the maximum torque capability of the motor is required.

The greatest benefits of flux optimisation occur when operating at low torque and speed. It is under these conditions that the greatest reduction in flux is possible and therefore the greatest reduction in iron core losses will occur. The new 'EV3' electric vehicle will be driven by a relatively powerful motor to give the vehicle high speed and acceleration. Under normal urban driving conditions, it will be operating at a relatively low torque compared to its full capability. Since the greatest efficiency increase is achieved at low torque, this vehicle has more to gain from flux magnitude optimisation that a vehicle driven by an induction motor that is operating near its torque limit during normal operation. Efficiency improvements of 2 to 6% have been shown in simulations of a normal driving cycle [Adnanes, Nilsen et al., 1993], with most savings being achieved in the low speed / low torque part of the driving cycle. The actual efficiency improvement due to flux magnitude optimisation is highly dependant on the actual motor / vehicle combination.

10. Conclusion

The University of Canterbury has been involved in the research and development of electric vehicles since 1974. These electric vehicles use induction motors that are designed with a low inductance such that they can run at high speed and therefore have a high power to weight ratio. The inverter switching can cause high current ripple due to the low inductance of these motors. A three-level inverter has been previously developed for electric vehicle applications that produces an improved voltage waveform and therefore has reduced current distortion.

The electric vehicles developed to date have been controlled using the slip-demand principle of induction motor torque control, but this control technique has poor performance. A number of high-performance induction motor torque control schemes have been investigated and four of these schemes were considered in detail. These were field oriented control, direct torque control, minimal torque ripple DTC, and DTC using space vector modulation. These four control schemes were simulated using MATLAB and Simulink.

The computer simulations showed that direct torque control and minimal torque ripple DTC had relatively high torque ripple, flux ripple, and current distortion during steady-state operation. Field oriented control and DTC using space vector modulation had better steady-state performance due to space vector modulation being used to synthesise the voltage. DTC using space vector modulation had some advantages over the traditional field oriented control scheme. Due to the absence of the PI current regulator, it had a significantly higher dynamic performance. It can also operate without an encoder, but this can degrade performance, particularly when operating at low speed.

DTC using space vector modulation was chosen as the torque control scheme for this application and its implementation was described in detail. Chapter 3 described the process for calculating the required voltage and chapter 4 described an implementation of space vector modulation for a three-level inverter. Chapter 4 also analysed the effect of the inverter dead-time. The inverter dead-time was found to introduce an error into the output voltage which causes an error in the stator flux estimated by the voltage model

stator flux observer. The dead-time can be compensated for by feeding back an estimate of the output voltage to the voltage model stator flux observer. The effect of the dead-time on the output voltage of a three-level inverter is significantly more time consuming to evaluate due to the increase number of inverter states.

DTC using space vector modulation was implemented and experimentally evaluated. A hardware platform for the controller was created that would be suitable for use in an electric vehicle. It was separated into a number of stackable PCBs to maximise its potential for reuse. At the base of this stack was a processor board based on a Microchip PIC16F877 microcontroller, a Texas Instruments TMS320VC33 DSP and a Xilinx XC4020XLA FPGA. An analogue expansion board was created to connect to the expansion bus that was on the processor board. The analogue expansion board contained a number of analogue-to-digital and digital-to-analogue conversion channels. Two analogue signal adapter boards were created that plugged into the analogue expansion board. These converted the motor stator currents, DC bus voltages, and the user torque reference to voltages suitable for the analogue-to-digital converter.

Four different components of software were created. The PIC microcontroller ran the System Manager software that had overall control of the processor board, and the DSP ran the Motor Control software that implemented the control algorithms. Another two software applications on a PC communicated with the processor board using RS232 serial connections. The Motor Manager software communicated with the System Manager software and was used to configure and control the system. The Motor Monitor software captured and displayed information output by the Motor Control software running on the DSP.

The implemented torque controller was experimentally evaluated using a previously constructed three-level IGBT inverter and a second-hand 15hp, 400Hz, 200V induction motor. This motor was later discovered to have a defective rotor, as it required 39% slip to obtain rated torque. Sufficient results were obtained to validate the performance of the torque controller despite the defective motor. When compared to two-level operation, the three-level inverter showed reduced current distortion when operating at modulation levels below 0.25 and above 0.9. The torque controller had a steady-state error in the torque produced due to errors in the stator flux estimate. This error was relatively

constant over the range of speeds that were tested. The estimated torque and stator flux responses to step changes in the reference values were almost instantaneous. A 5Nm change in torque (55% of rated torque) occurred in 200μs and a 0.02wb change in stator flux (43% of rated flux) occurred in 300μs.

A number of improvements to the torque controller were suggested. Improving the stator flux estimation would increase the torque accuracy. The response to large changes in the torque and flux references could be improved using one of the techniques described in Section 3.2.5. It may also be possible to improve the efficiency of the system using flux magnitude optimisation. For every motor speed and torque there will be a flux level that will result in maximum efficiency.

In summary, a three-level IGBT inverter has been previously developed for an electric vehicle application. Compared to a standard two-level inverter, this new inverter requires more complex switching patterns but can generate improved voltage waveforms. This is an advantage as it reduces the current distortion when driving the high-speed, low inductance motors that are used in electric vehicles. Four different induction motor torque control schemes suitable for an electric vehicle application using a three-level inverter were evaluated using computer simulations. A direct torque control using space vector modulation control scheme was chosen based on its good steady state and dynamic performance. This torque control scheme was implemented on a custom-built embedded controller and tested against a 15hp, 400Hz, 200V induction motor. A steady-state torque error was observed due to errors in the stator flux estimation, but the torque and flux responded almost instantaneously to step changes in the reference values, which matched the simulation results. This project has implemented a torque controller that could be successfully used with a three-level inverter in an electric vehicle application.

References

Adnanes, A. K., Nilsen, R., Loken, R. and Norum, L. (1993), "Efficiency analysis of electric vehicles, with emphasis on efficiency optimized excitation", *Industry Applications Society Annual Meeting, 1993., Conference Record of the 1993 IEEE*, Vol. 1, Oct. 1993, pp 455-462.

Anton, H. (1994), "Elementary linear algebra", John Wiley, New York, pp 101-110.

Attaianese, C., Nardt, V., Perfetto, A. and Tomasso, G. (1999), "Vectorial torque control: a novel approach to torque and flux control of induction motor drives", *Industry Applications, IEEE Transactions on*, Vol. 35, No. 6, Nov/Dec 1999, pp 1399-1405.

Bellini, A., Filippetti, F., Franceschini, G., Tassoni, C. and Kliman, G. B. (2001), "Quantitative evaluation of induction motor broken bars by means of electrical signature analysis", *Industry Applications, IEEE Transactions on*, Vol. 37, No. 5, Sept/Oct 2001, pp 1248-1255.

Boys, J. T. and Handley, P. G. (1990), "Harmonic analysis of space vector modulated PWM waveforms", *Electric Power Applications, IEE Proceedings*, Vol. 137, No. 4, July 1990, pp 197-204.

Byers, D. J. (1983), "AC Induction motor drive systems for electric vehicles", *The Institution of Professional Engineers New Zealand (IPENZ), Transactions of*, Vol. 10, 1983, pp 1-14.

Carrara, G., Gardella, S., Marchesoni, M., Salutari, R. and Sciutto, G. (1992), "A new multilevel PWM method: a theoretical analysis", *Power Electronics, IEEE Transactions on*, Vol. 7, No. 3, July 1992, pp 497-505.

Casadei, D., Grandi, G., Serra, G. and Tani, A. (1994), "Effects of flux and torque hysteresis band amplitude in direct torque control of induction machines",

Industrial Electronics, Control and Instrumentation, 1994. IECON '94., 20th International Conference on, Vol. 1, Sept. 1994, pp 299-304.

Casadei, D., Serra, G. and Tani, K. (2000), "Implementation of a direct control algorithm for induction motors based on discrete space vector modulation", *Power Electronics, IEEE Transactions on*, Vol. 15, No. 4, July 2000, pp 769-777.

Depenbrock, M. (1988), "Direct self-control of inverter-fed machine", *Power Electronics, IEEE Transactions on*, Vol. 3, No. 4, Oct. 1988, pp 420-429.

Griva, G., Habetler, T. G., Profumo, F. and Pastorelli, M. (1995), "Performance evaluation of a direct torque controlled drive in the continuous PWM-square wave transition region", *Power Electronics, IEEE Transactions on*, Vol. 10, No. 4, July 1995, pp 464-471.

Habetler, T. G., Profumo, F., Griva, G., Pastorelli, M. and Bettini, A. (1998), "Stator resistance tuning in a stator-flux field-oriented drive using an instantaneous hybrid flux estimator", *Power Electronics, IEEE Transactions on*, Vol. 13, No. 1, Jan 1998, pp 125 - 133.

Habetler, T. G., Profumo, F., Pastorelli, M. and Tolbert, L. M. (1991), "Direct torque control of induction machines using space vector modulation", *Industry Applications Society Annual Meeting, 1991., Conference Record of the 1991 IEEE*, Vol. 1, Sept/Oct 1991, pp 428-436.

Harman, R. T. C. (1992), "Experience with electric vehicle concepts and operation", *Urban Electric Vehicle., International Conference on the*, May 1992.

Holtz, J. (1994), "Pulsewidth modulation for electronic power conversion", *Proceedings of the IEEE*, Vol. 82, No. 8, Aug 1994, pp 1194-1214.

Jansen, P. L. and Lorenz, R. D. (1994), "A physically insightful approach to the design and accuracy assessment of flux observers for field oriented induction machine

drives", *Industry Applications, IEEE Transactions on*, Vol. 30, No. 1, Jan/Feb 1994, pp 101-110.

Jeong, S.-G., Lee, B.-S., Kim, K.-S. and Park, M.-H. (1988), "The Analysis and Compensation of Dead Time Effects in PWM Inverters", *Industrial Electronics Society, 1988. IECON '88. Proceedings., 14 Annual Conference of*, Vol. 3, Oct. 1988, pp 667-671.

Kang, J.-K. and Sul, S.-K. (1999), "New direct torque control of induction motor for minimum torque ripple and constant switching frequency", *Industry Applications, IEEE Transactions on*, Vol. 35, No. 5, Sept/Oct 1999, pp 1076-1082.

Lascu, C., Boldea, I. and Blaabjerg, F. (1998), "A modified direct torque control (DTC) for induction motor sensorless drive", *Industry Applications Conference, 1998. Thirty-Third IAS Annual Meeting. The 1998 IEEE*, Vol. 1, Oct. 1998, pp 415-422.

Lee, Y.-H., Suh, B.-S. and Hyun, D.-S. (1996), "A novel PWM scheme for a three-level voltage source inverter with GTO thyristors", *Industry Applications, IEEE Transactions on*, Vol. 32, No. 2, Mar/Apr 1996, pp 260-268.

Li, X. (1999), "Design of a three-level inverter for an electric vehicle", Masters Thesis, *Electrical and Electronic Engineering Dept., University of Canterbury*, 112 pages.

Liu, H. L. and Cho, G. H. (1994), "Three-level space vector PWM in low index modulation region avoiding narrow pulse problem", *Power Electronics, IEEE Transactions on*, Vol. 9, No. 5, Sept. 1994, pp 481-486.

Liu, H. L., Choi, N. S. and Cho, G. H. (1991), "DSP based space vector PWM for three-level inverter with DC-link voltage balancing", *Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON '91., 1991 International Conference on*, Oct 1991, pp 197-203.

Liu, Y.-H. and Chen, C.-L. (1998), "Novel dead time compensation method for induction motor drives using space vector modulation", *Electric Power Applications, IEE Proceedings*, Vol. 145, No. 4, July 1998, pp 387-392.

Manjrekar, M. D. and Venkataramanan, G. (1996), "Advanced topologies and modulation strategies for multilevel inverters", *Power Electronics Specialists Conference, 1996. PESC '96 Record., 27th Annual IEEE*, 1996, pp 1013 -1018.

Martins, C. A., Carvalho, A. S., Roboam, X. and Meynard, T. A. (1998), "Evolution of induction motor control methods and related voltage-source inverter topologies", *Advanced Motion Control, 1998. AMC '98-Coimbra., 1998 5th International Workshop on*, Jun. 1998, pp 15-20.

Matsuse, K., Taniguchi, S., Yoshizumi, T. and Namiki, K. (2001), "A speed-sensorless vector control of induction motor operating at high efficiency taking core loss into account", *Industry Applications, IEEE Transactions on*, Vol. 37, No. 2, March-April 2001, pp 548-558.

Murai, Y., Watanabe, T. and Iwasaki, H. (1987), "Waveform distortion and correction circuit for PWM inverters with switching lag-times", *Industry Applications, IEEE Transactions on*, Vol. 23, Sept/Oct 1987, pp 881-886.

Nabae, A., Takahashi, I. and Akagi, H. (1981), "A new neutral-point-clamped PWM inverter", *Industry Applications, IEEE Transactions on*, Vol. 17, No. 5, Sept/Oct 1981, pp 518-523.

Novotny, D. W. and Lipo, T. A. (1997), "Vector control and dynamics of AC drives", Oxford University Press, New York

Oh, W. S., Kim, Y. T. and Kim, H. J. (1995), "Dead time compensation of current controlled inverter using space vector modulation method", *Power Electronics and Drive Systems, 1995., Proceedings of 1995 International Conference on*, Vol. 1, Feb. 1995, pp 374-378.

Seo, J. H., Choi, C. H. and Hyun, D. S. (2001), "A new simplified space-vector PWM method for three-level inverters", *Power Electronics, IEEE Transactions on*, Vol. 16, No. 4, July 2001, pp 545-550.

Steinke, J. K. (1989), "Switching frequency optimal PWM control of a three-level inverter", *Power Electronics, IEEE Transactions on*, Vol. 7, No. 3, July 1992, pp 487-496.

Suh, B.-S., Sinha, G., Manjrekar, M. D. and Lipo, T. A. (1998), "Multilevel Power Conversion - An Overview Of Topologies And Modulation Strategies", *Optimization of Electrical and Electronic Equipments, 1998. OPTIM '98. Proceedings of the 6th International Conference on*, May 1998, pp AD-11 to AD-24.

Takahashi, I. and Noguchi, T. (1986), "A new quick-response and high-efficiency control strategy of an induction motor", *Industry Applications, IEEE Transactions on*, Vol. 22, No. 5, Sept/Oct. 1986, pp 820-827.

Takahashi, I. and Ohmori, Y. (1989), "High-performance direct torque control of an induction motor", *Industry Applications, IEEE Transactions on*, Vol. 25, No. 2, Mar/Apr 1989, pp 257-264.

Tolbert, L. M., Peng, F. Z. and Habetler, T. G. (1998), "Multilevel inverters for electric vehicle applications", *Power Electronics in Transportation, 1998*, Oct. 1998, pp 79-84.

van der Broeck, H. W., Skudelny, H.-C. and Stanke, G. V. (1988), "Analysis and Realization of a Pulsewidth Modulator Based on Voltage Space Vectors", *Industry Applications, IEEE Transactions on*, Vol. 24, No. 1, Jan/Feb 1998, pp 142-150.

Williams, M. R. (1993), "A current controlled inverter for an electric vehicle", Masters Thesis, *Electrical and Electronic Engineering Dept., University of Canterbury*, 147 pages.

Xue, Y., Xu, X., Habetler, T. G. and Divan, D. M. (1990), "A low cost stator flux oriented voltage source variable speed drive", *Industry Applications Society Annual Meeting, 1990., Conference Record of the 1990 IEEE*, Vol. 1, Oct. 1990, pp 410-415.

Zhang, J. (1995), "High performance control of a three-level IGBT inverter fed AC drive", *Industry Applications Conference, 1995. Thirtieth IAS Annual Meeting, IAS '95., Conference Record of the 1995 IEEE*, Oct. 1995, pp 22-28 vol.1.

Appendix A. Simulink Models

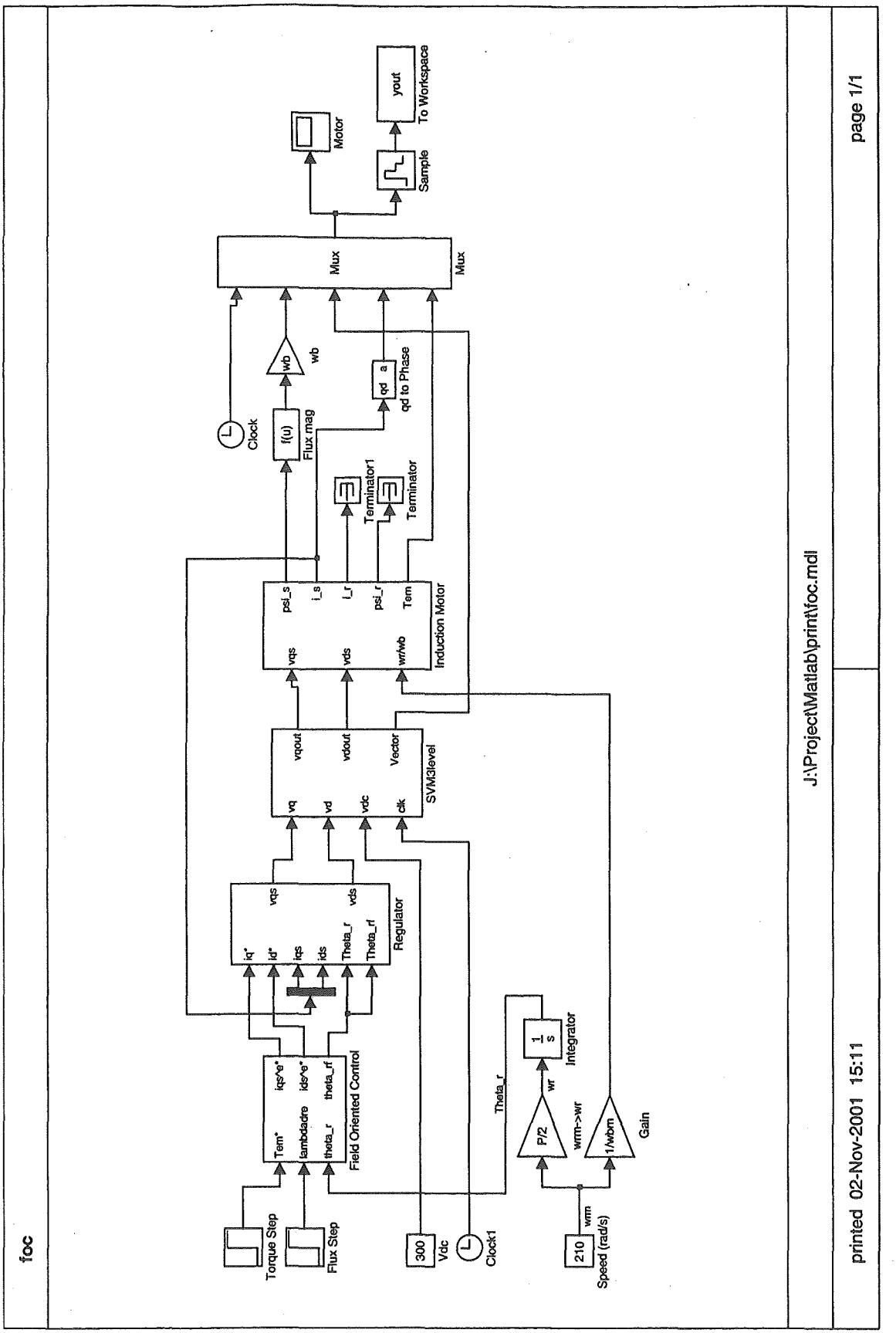


Figure A.1: Field Oriented Controller Model - Top Level

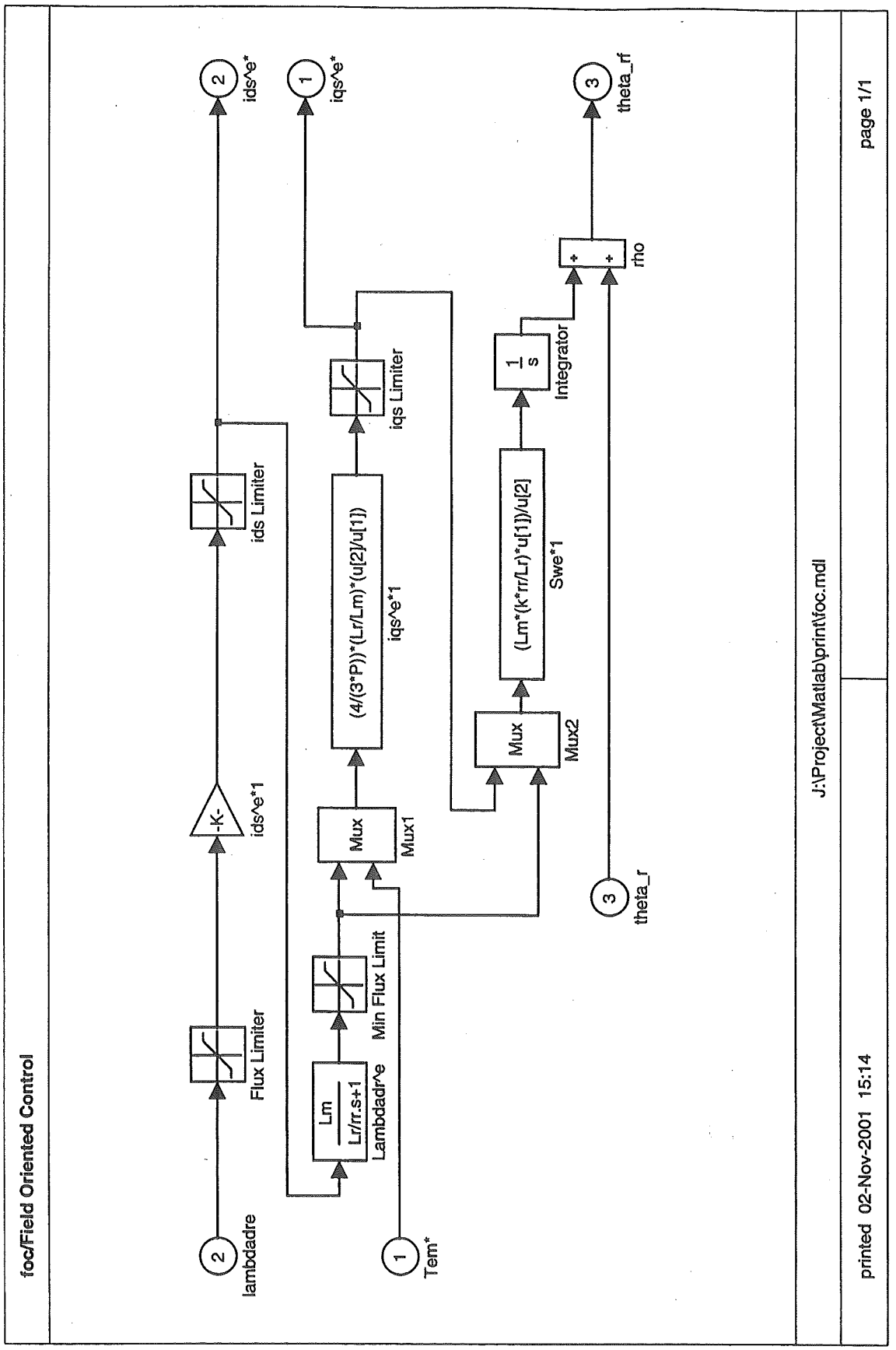


Figure A.2: Field Oriented Controller Model - FOC Block

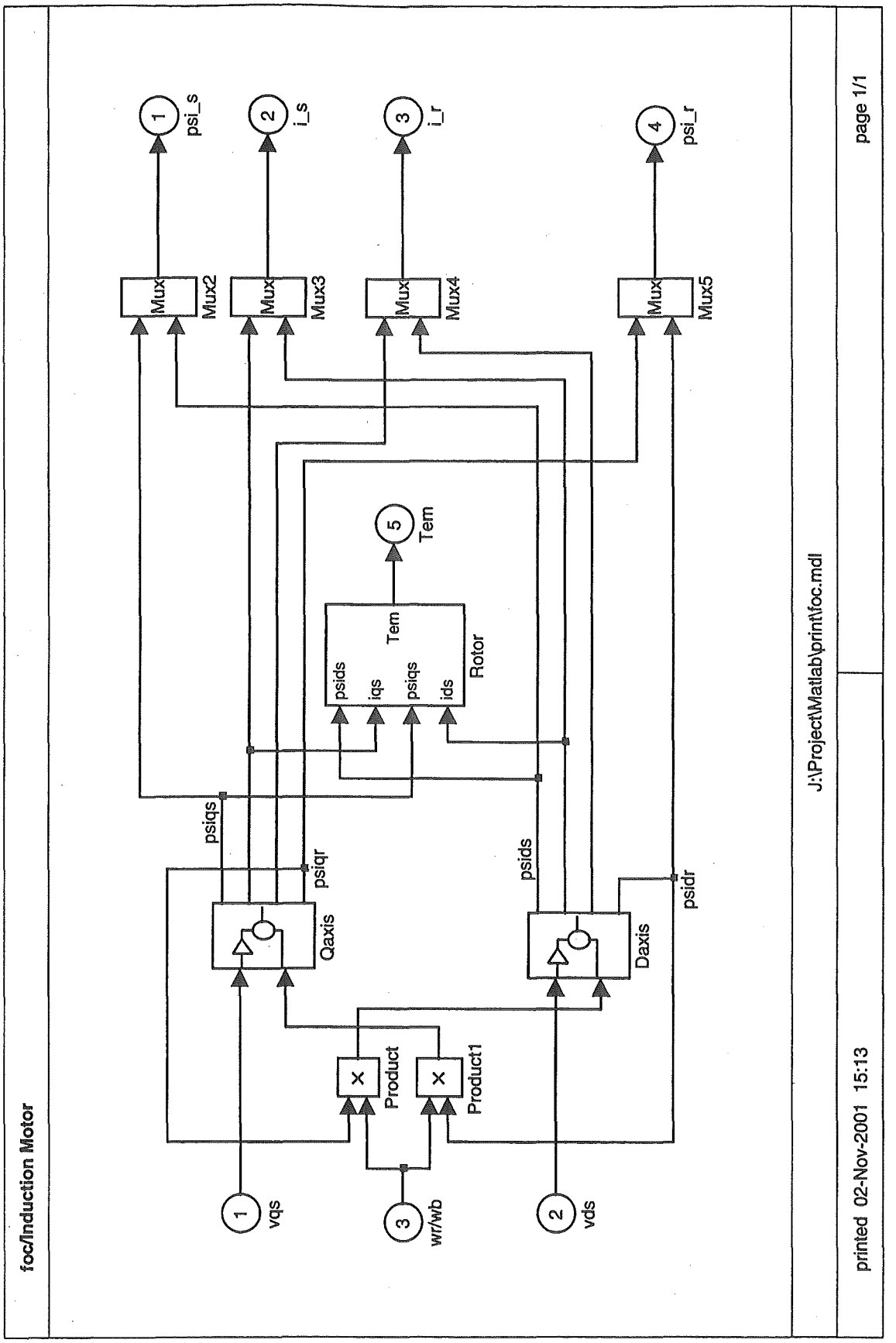


Figure A.3: Field Oriented Controller Model - Induction Motor Block

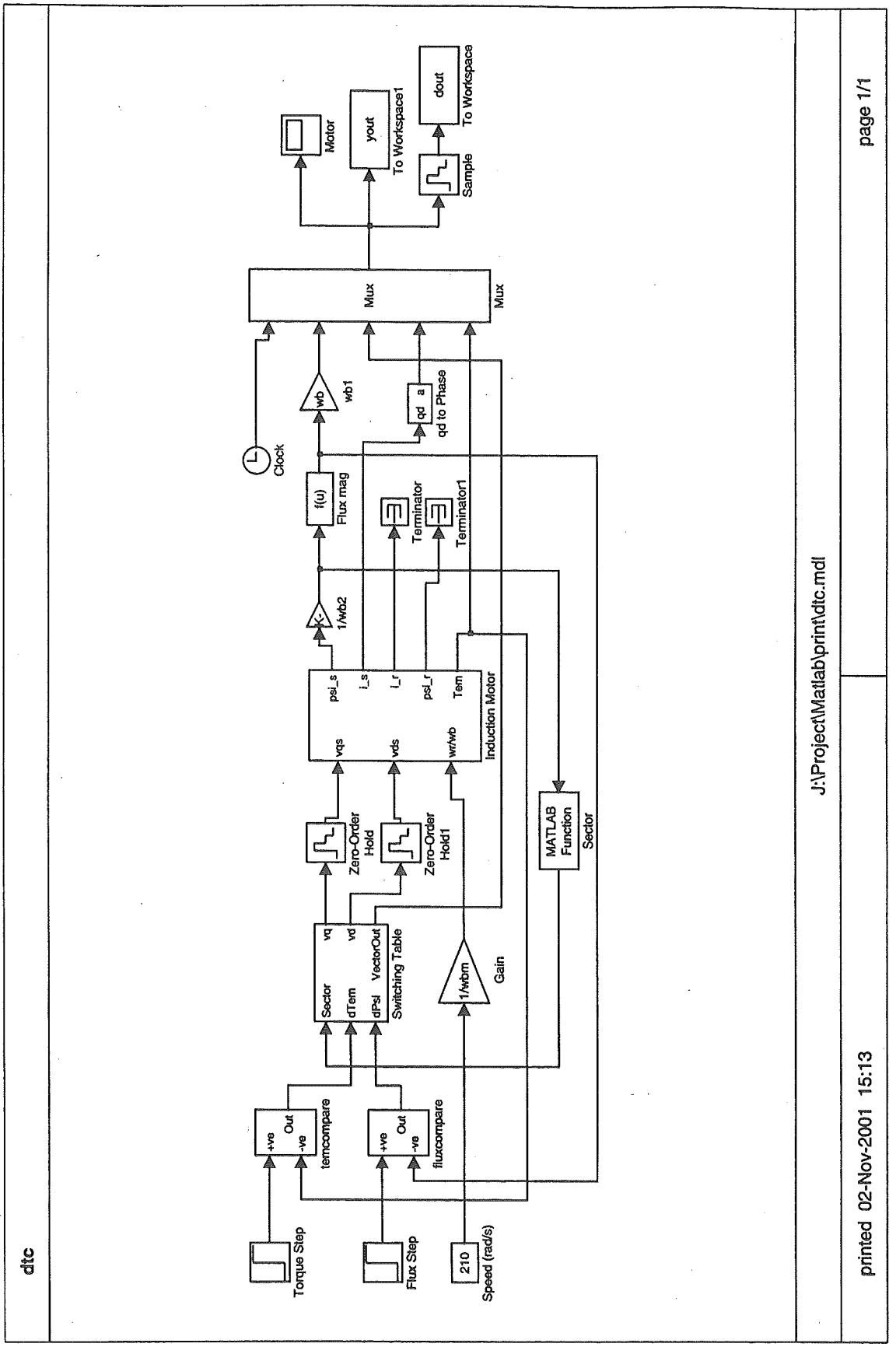


Figure A.4: Direct Torque Controller Model

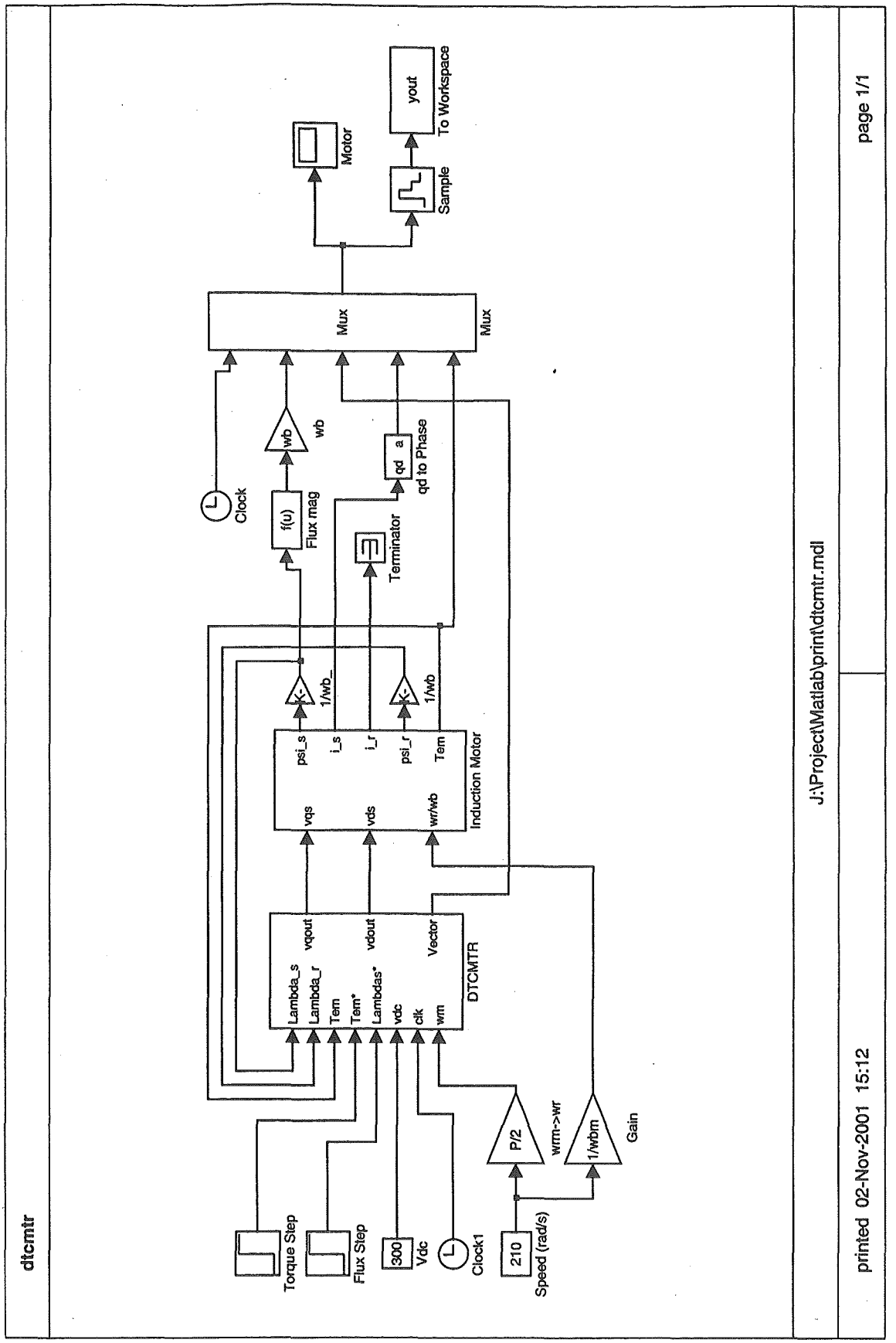


Figure A.5: Minimal Torque Ripple DTC Model

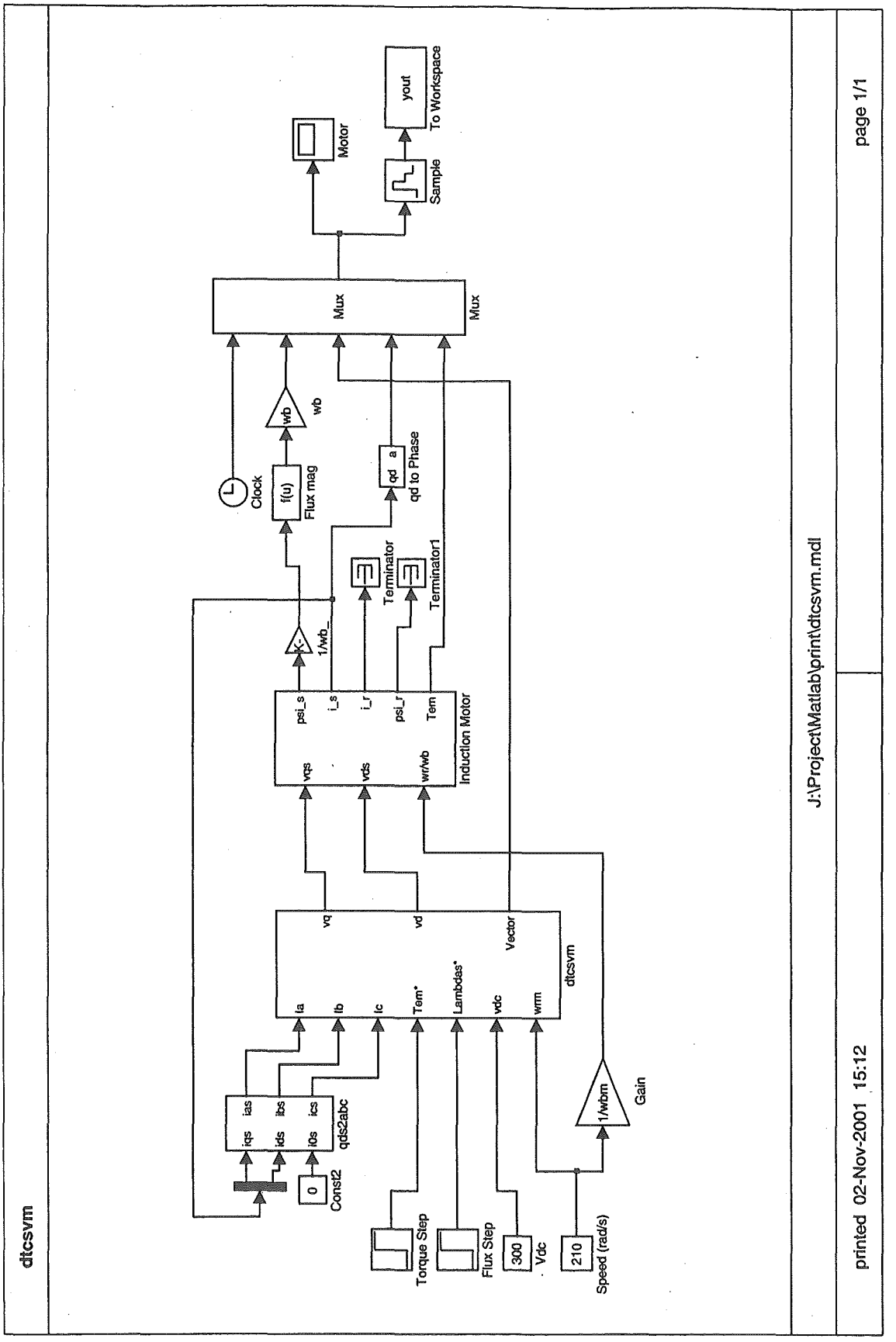


Figure A.6: DTC using Space Vector Modulation Model

Appendix B. Circuit Schematics

B-1. Processor Board

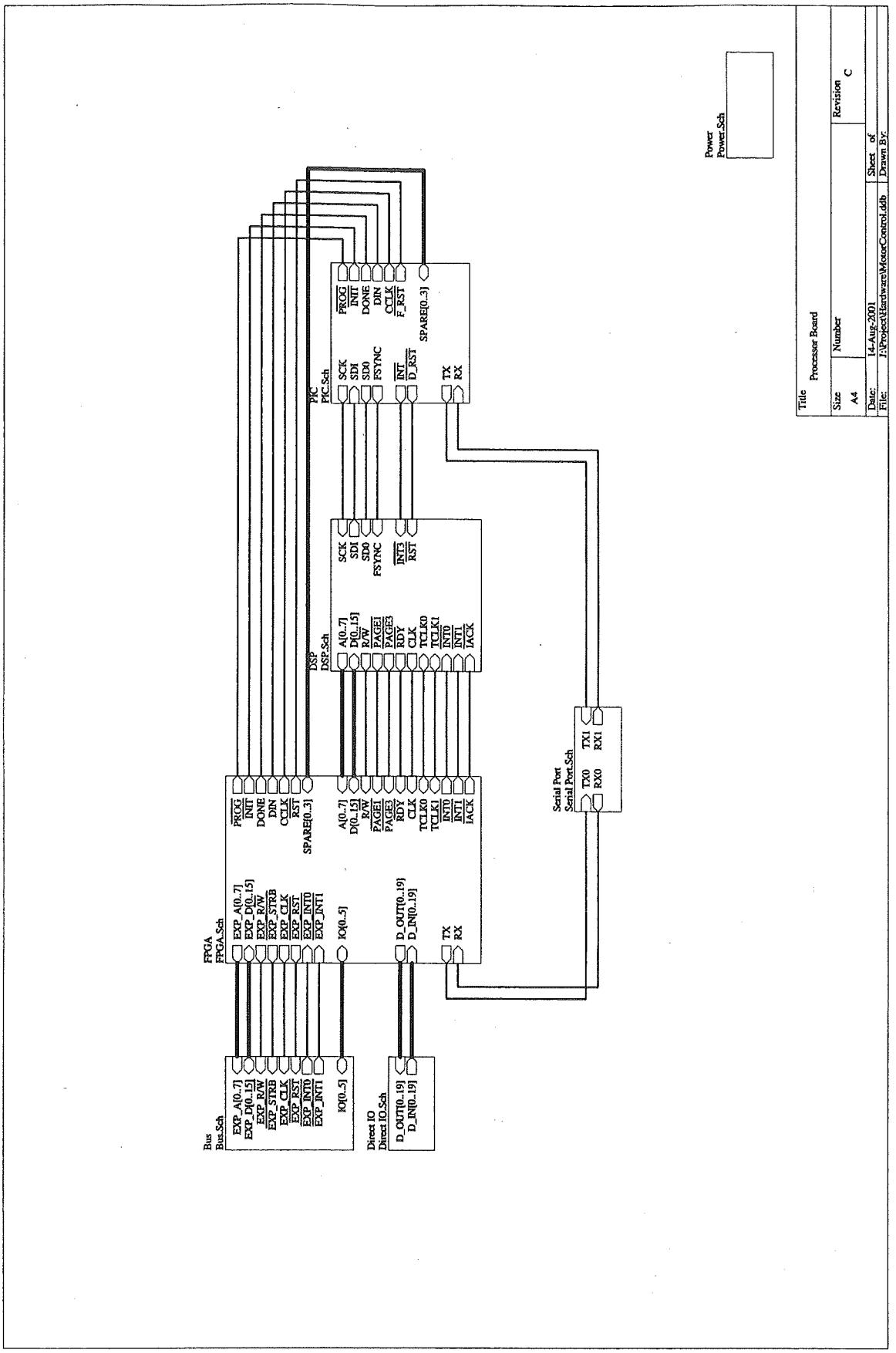


Figure B1.1: Processor Board Schematic - Top Level

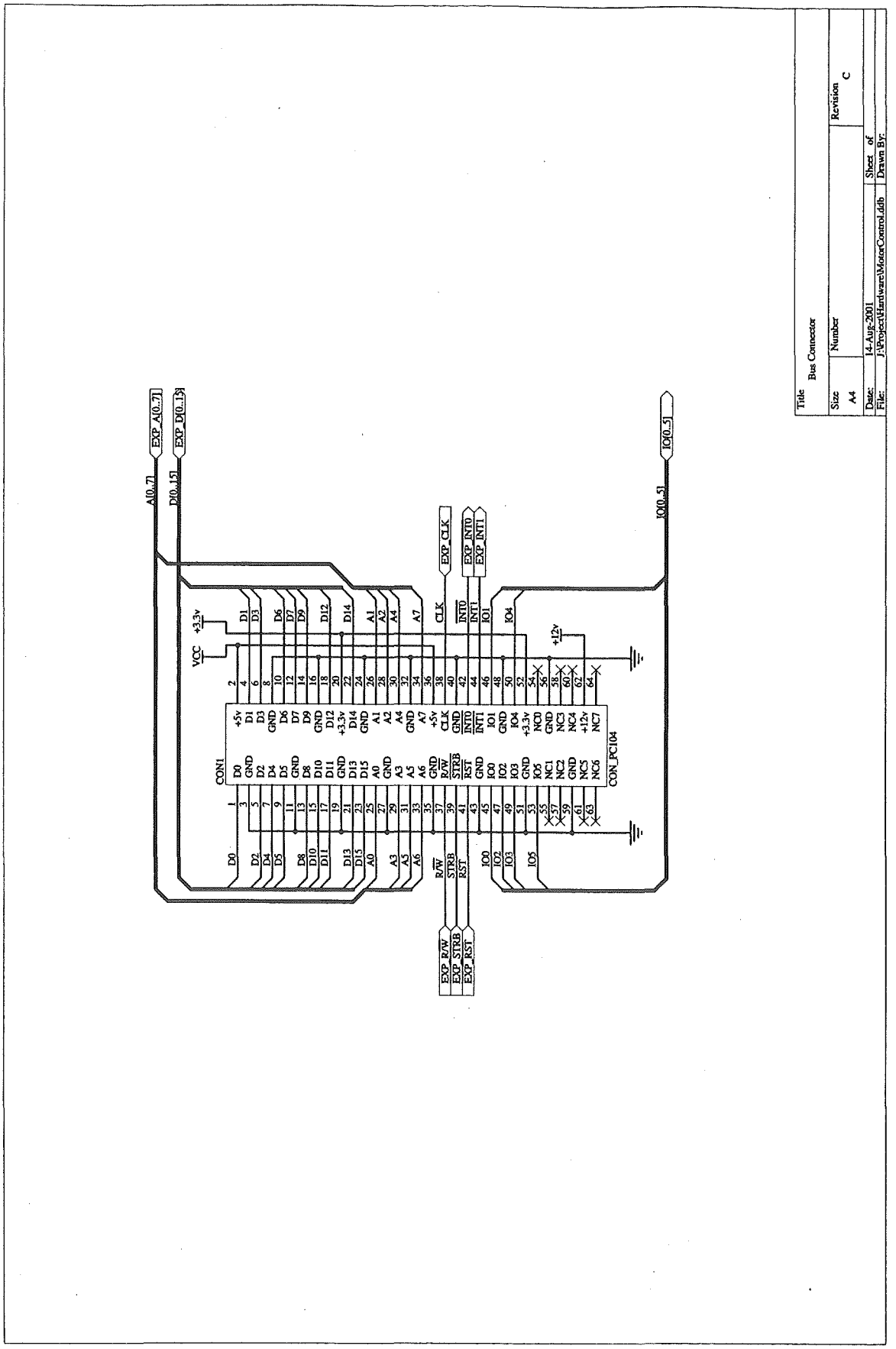


Figure B1.2: Processor Board Schematic - Expansion Bus

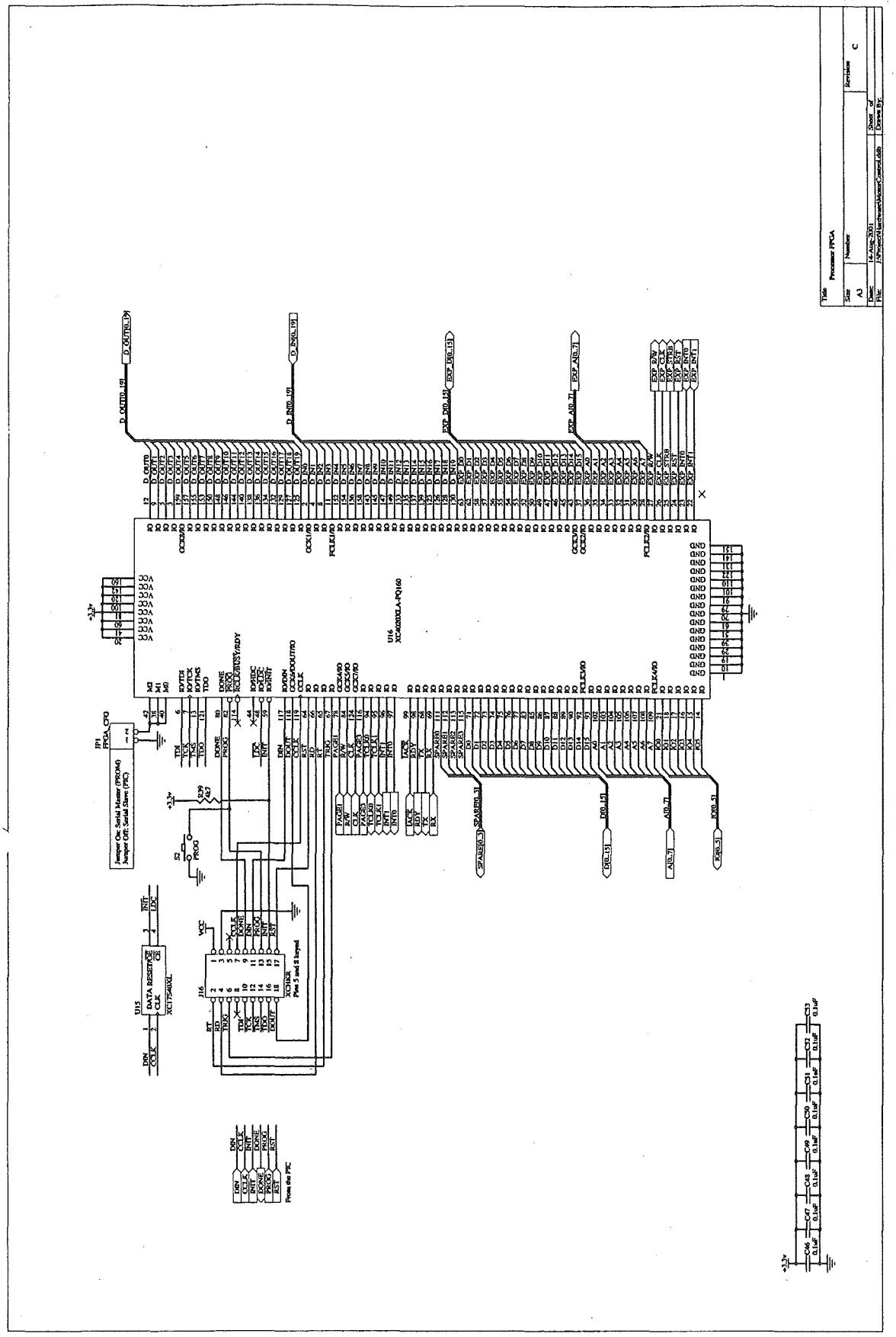


Figure B1.3: Processor Board Schematic - FPGA

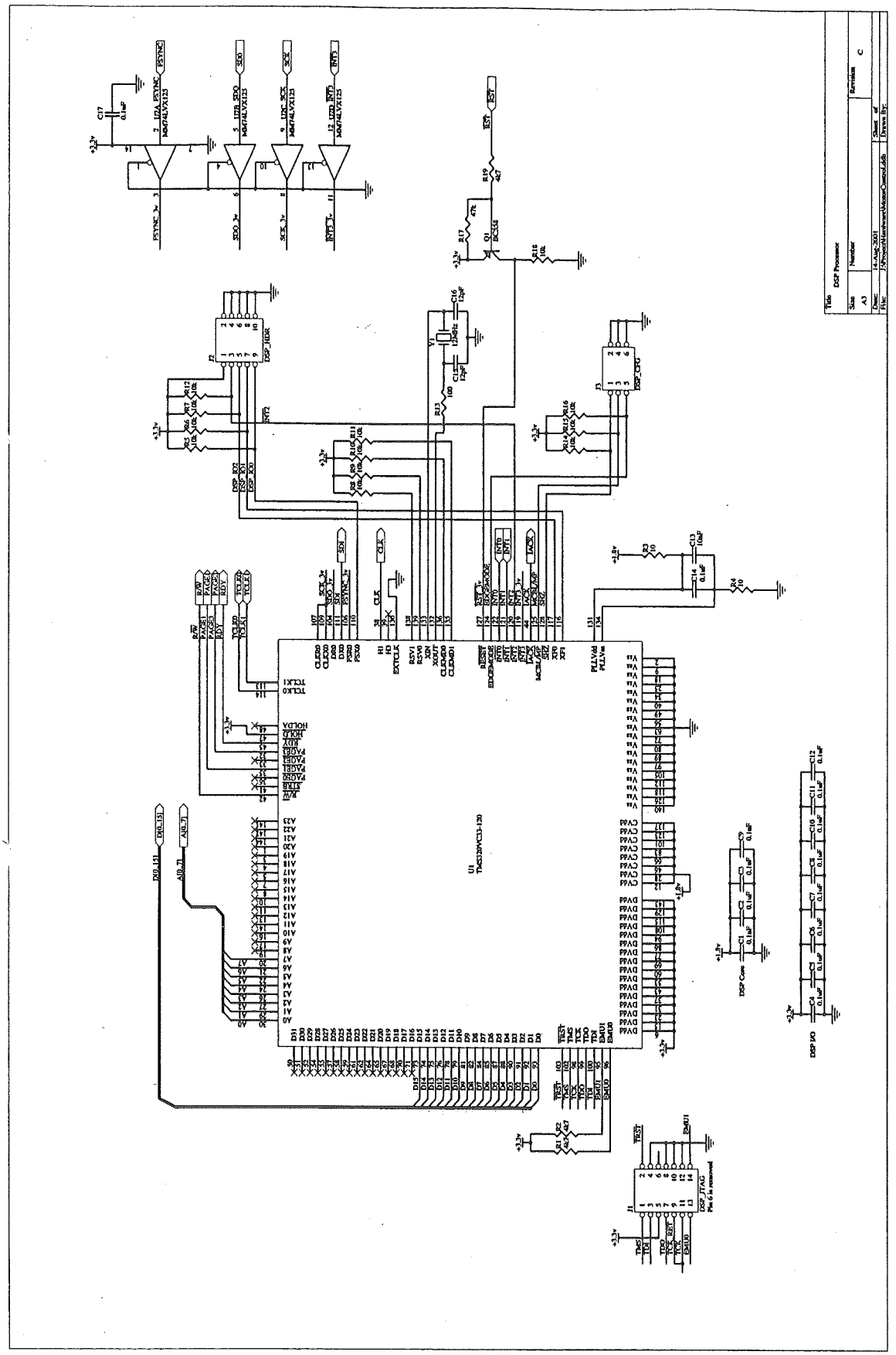


Figure B1.4: Processor Board Schematic - DSP

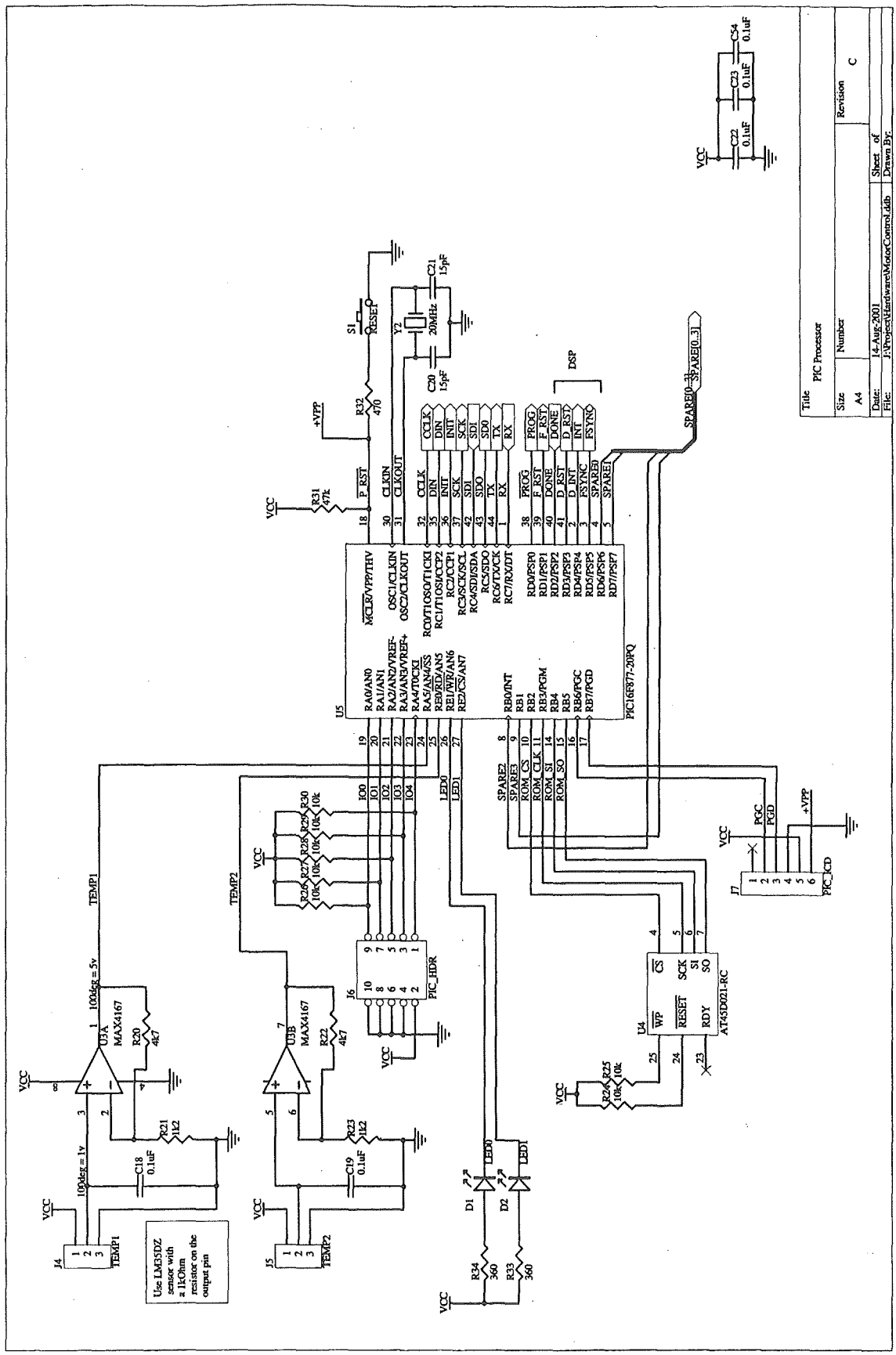


Figure B1.5: Processor Board Schematic - PIC Microcontroller

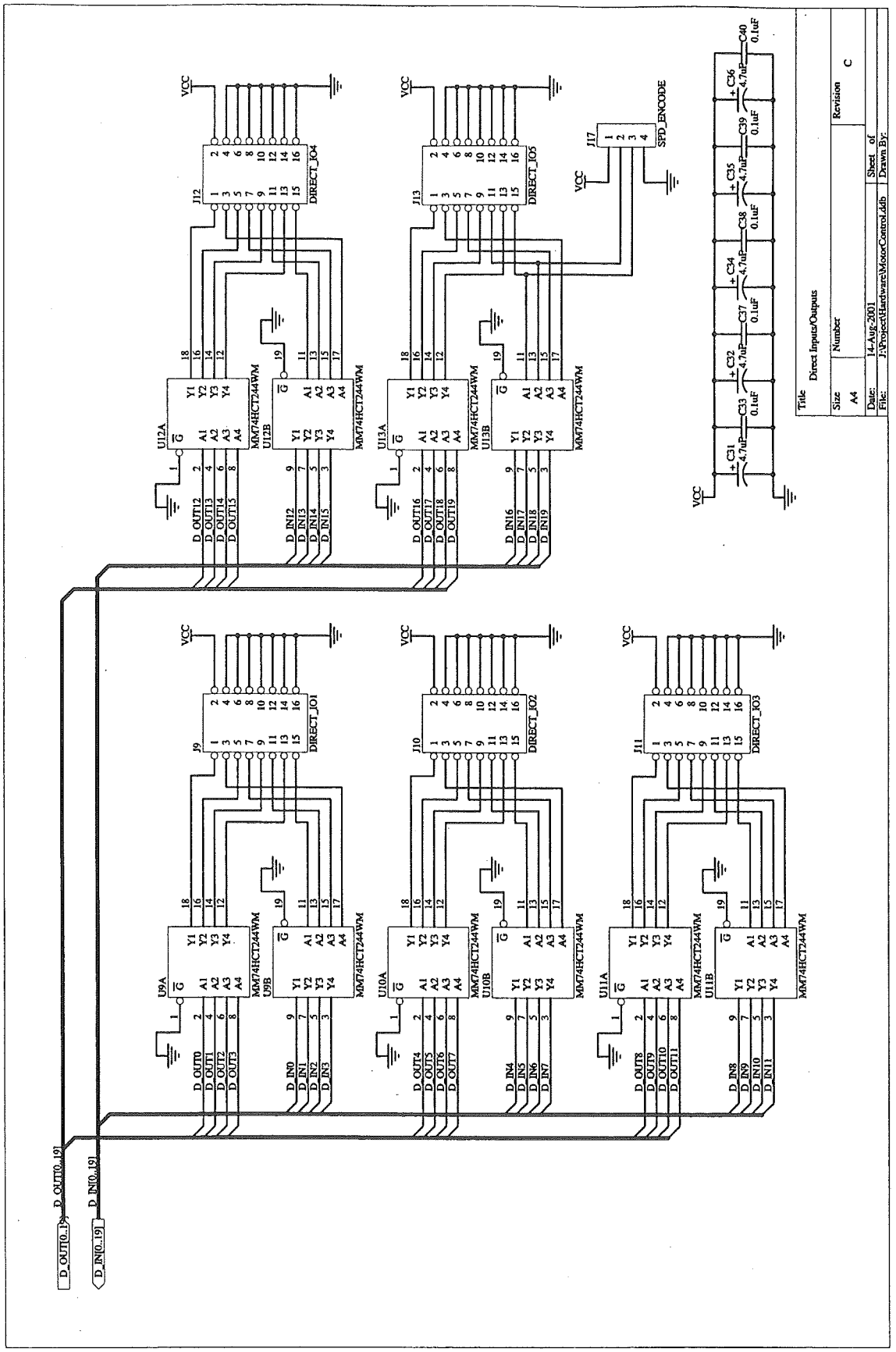


Figure B1.6: Processor Board Schematic - Digital IO

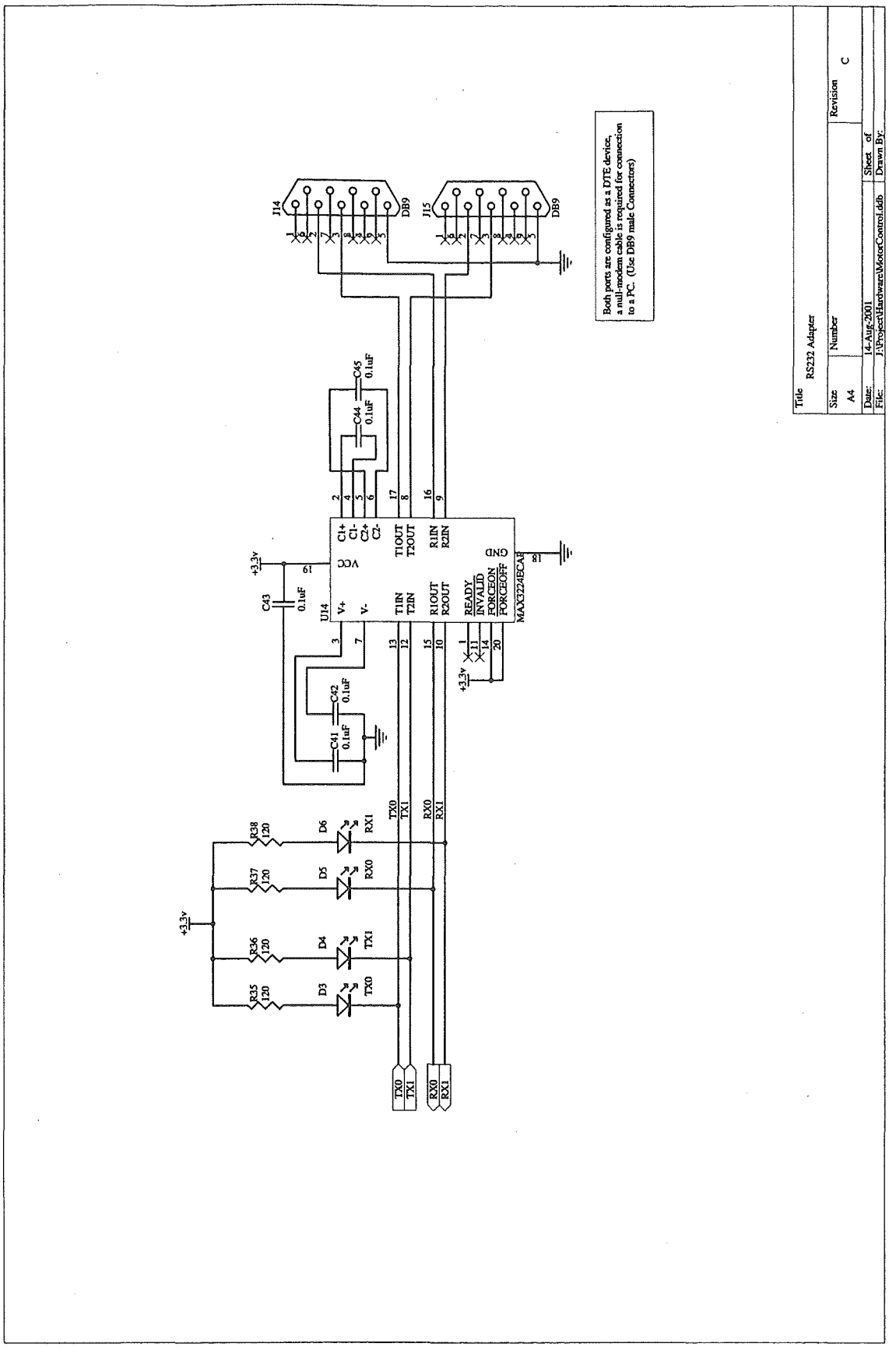


Figure B1.7: Processor Board Schematic - RS232 Transceiver

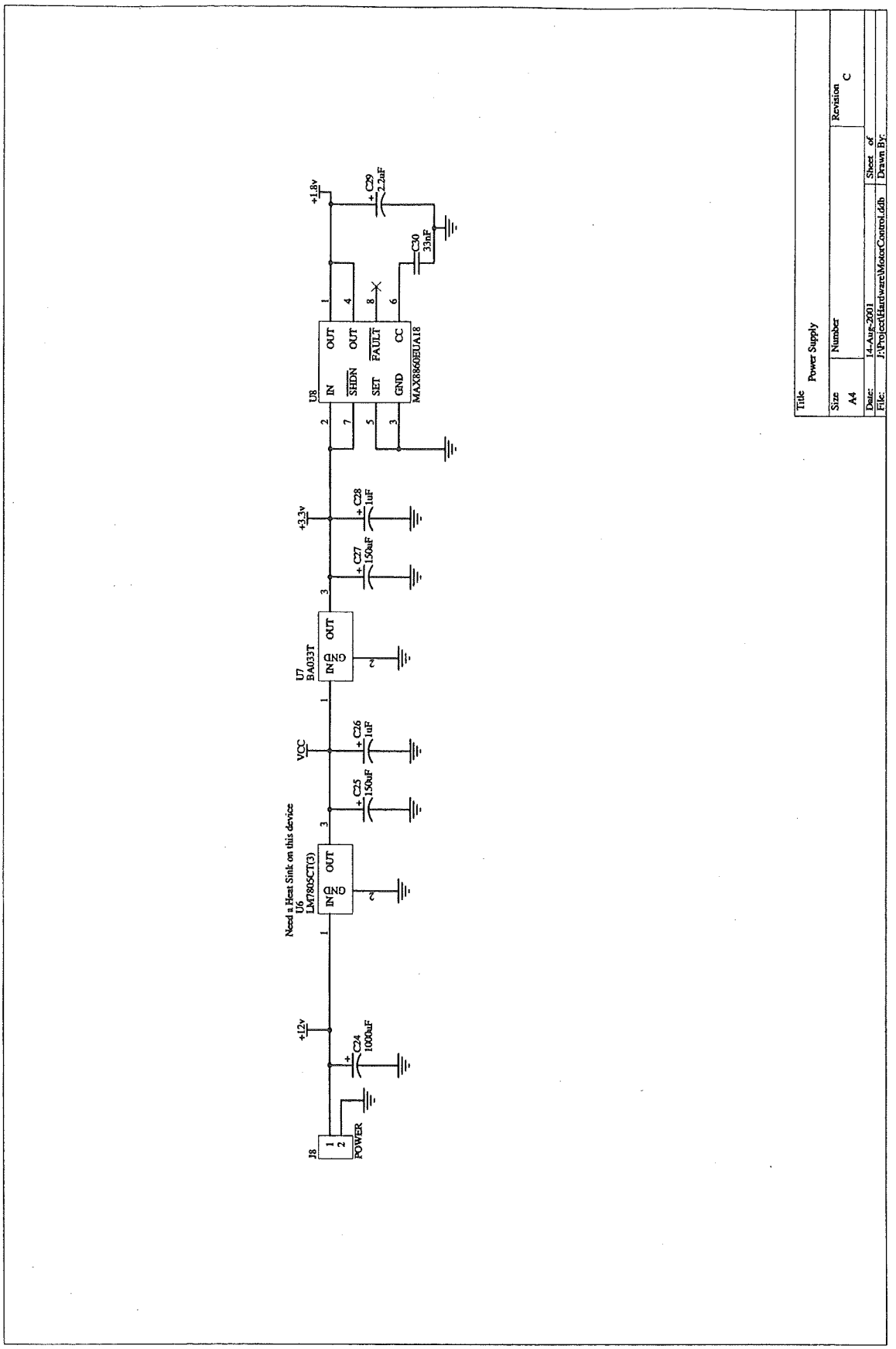


Figure B1.8: Processor Board Schematic - Power Supplies

B-2. Analogue Expansion Board

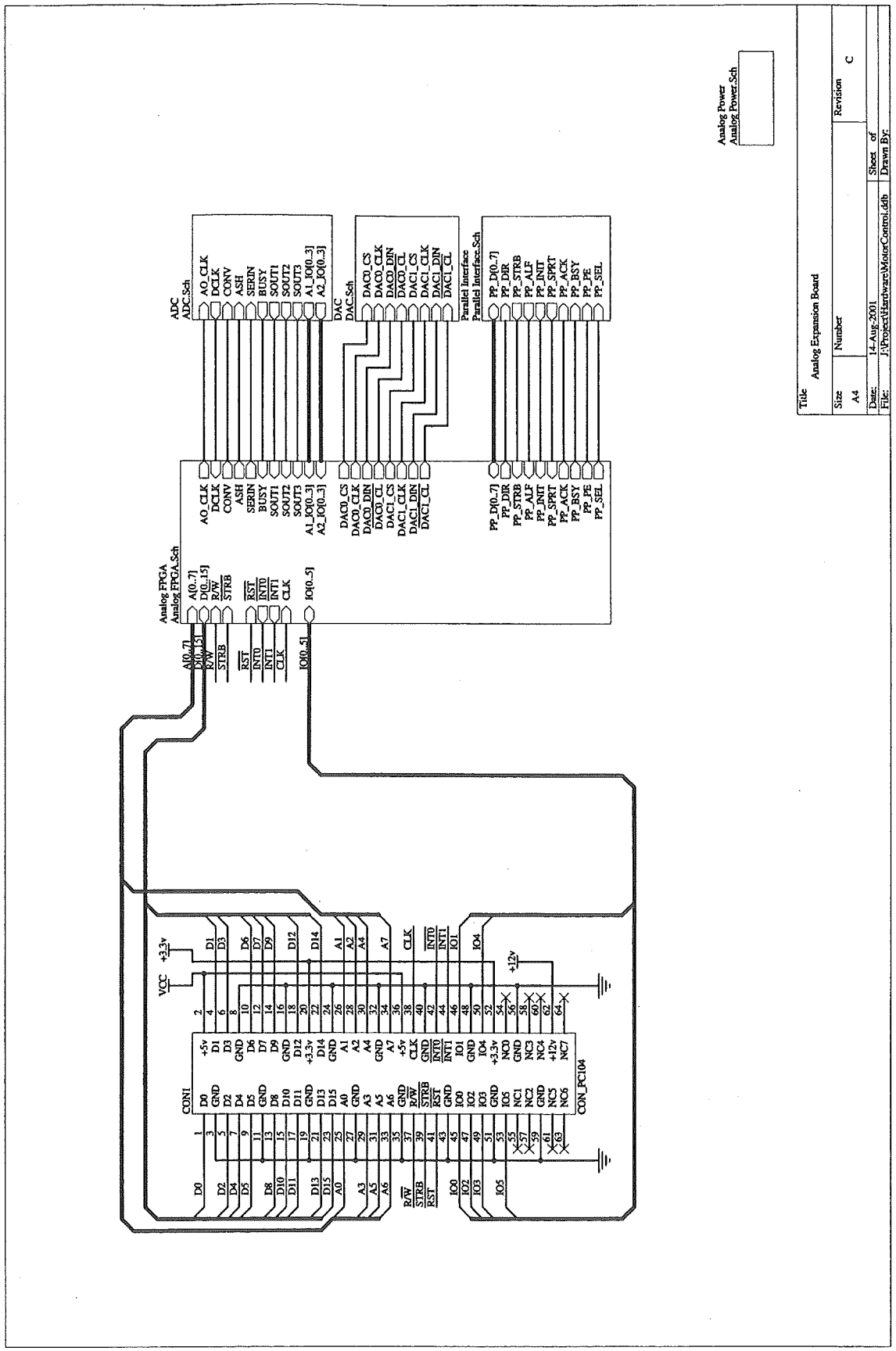


Figure B2.1: Analogue Expansion Board Schematic - Top Level

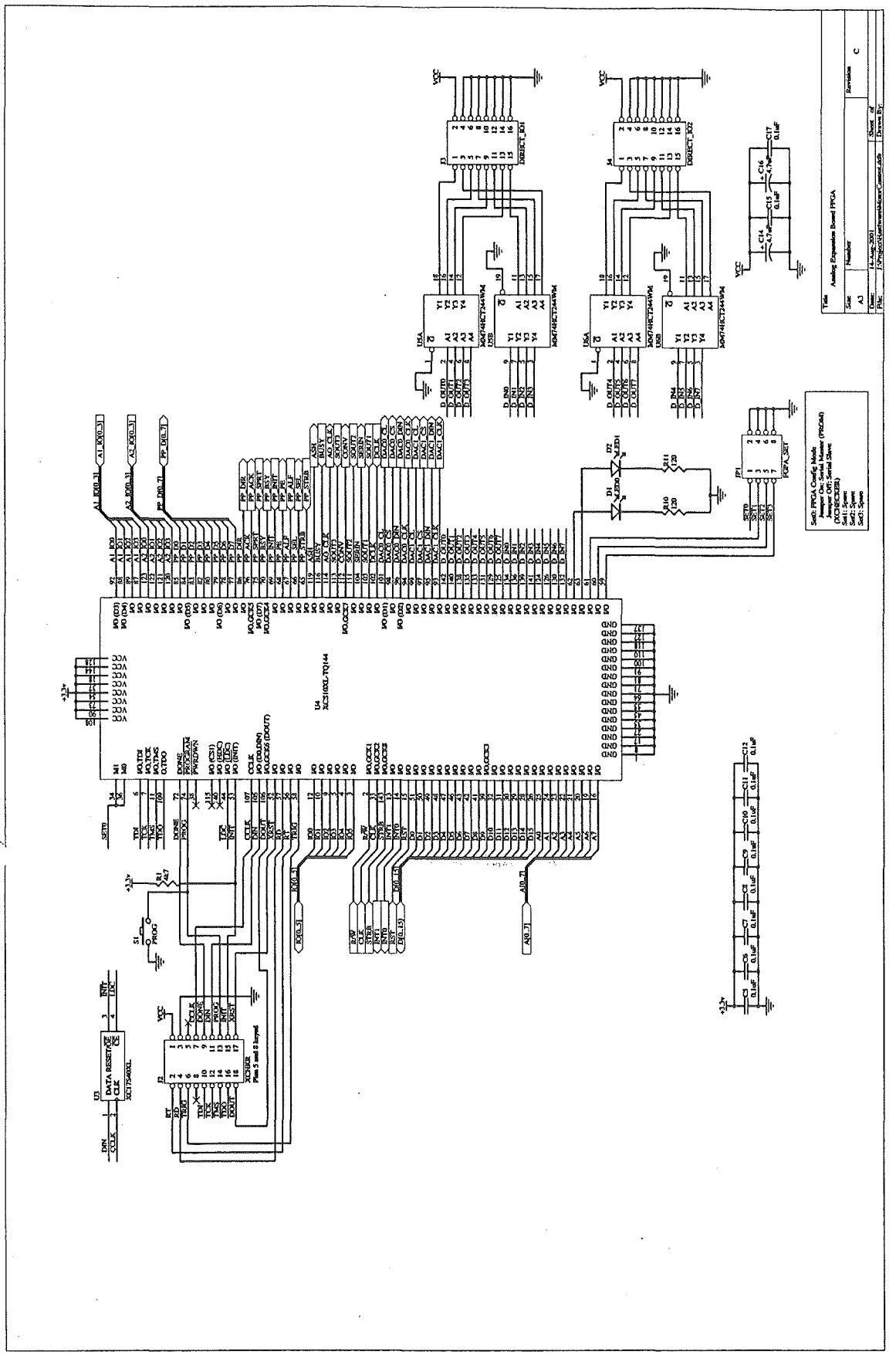


Figure B2.2: Analogue Expansion Board Schematic - FPGA

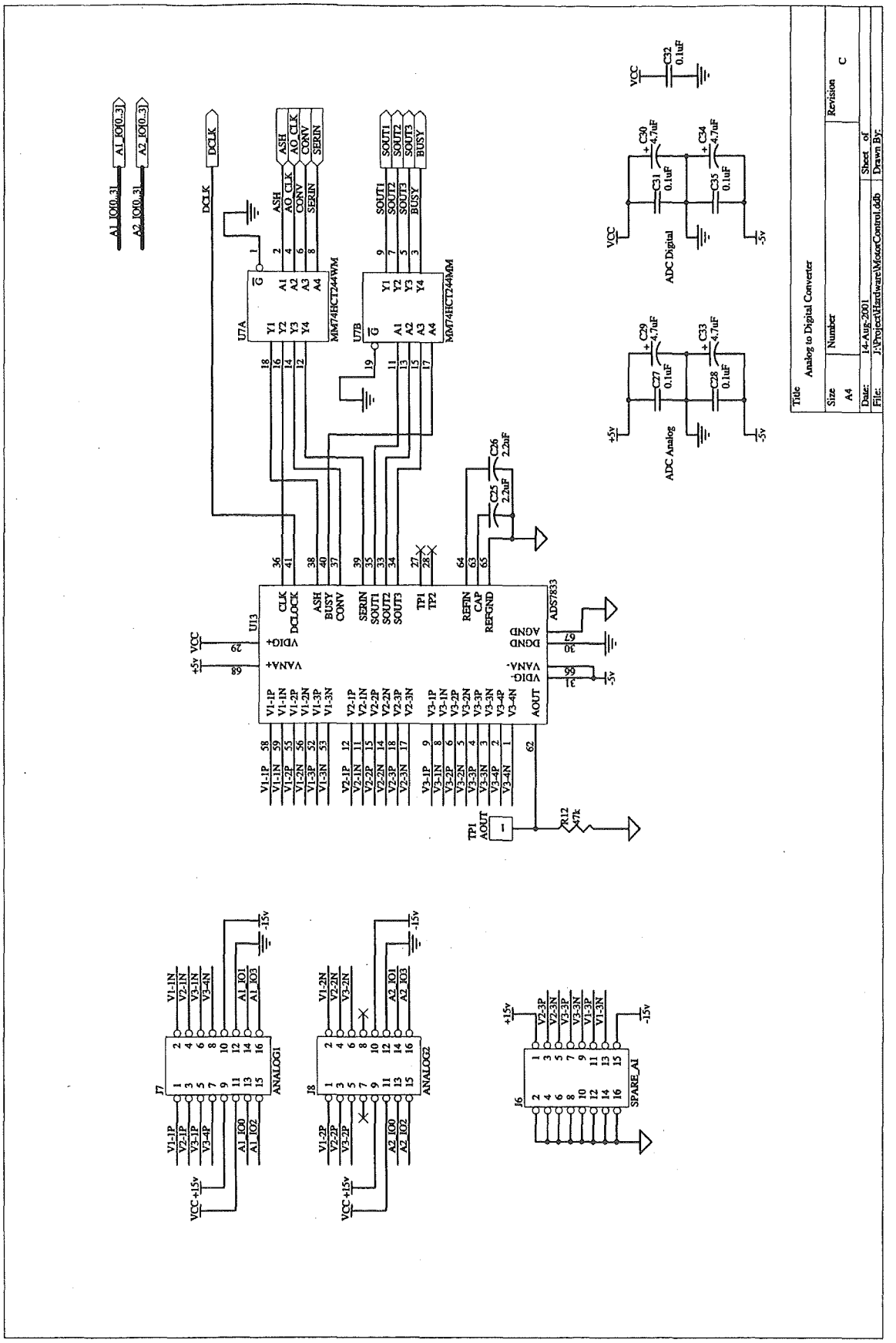


Figure B2.3: Analogue Expansion Board Schematic - ADC

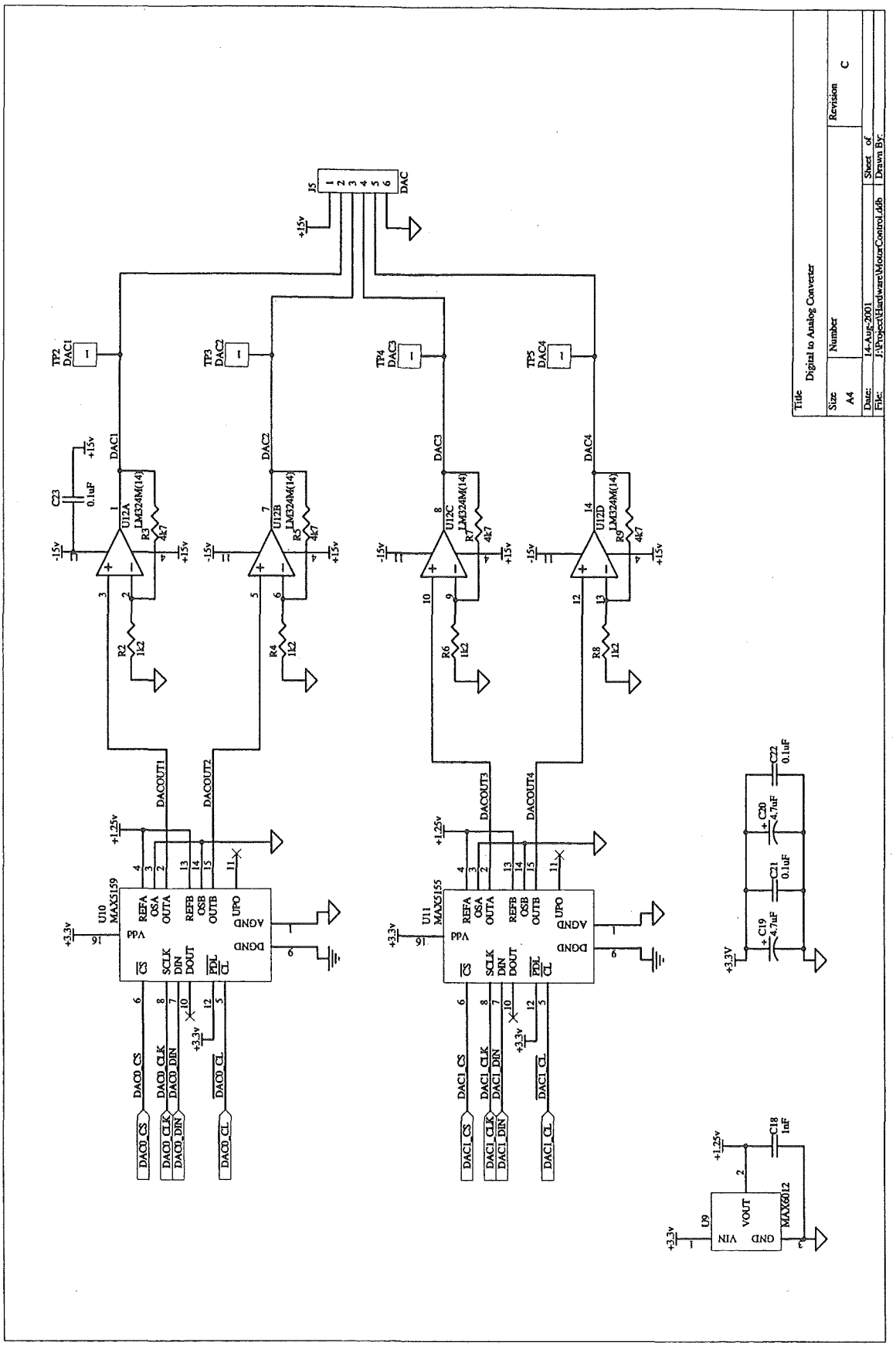


Figure B2.4: Analogue Expansion Board Schematic - DAC

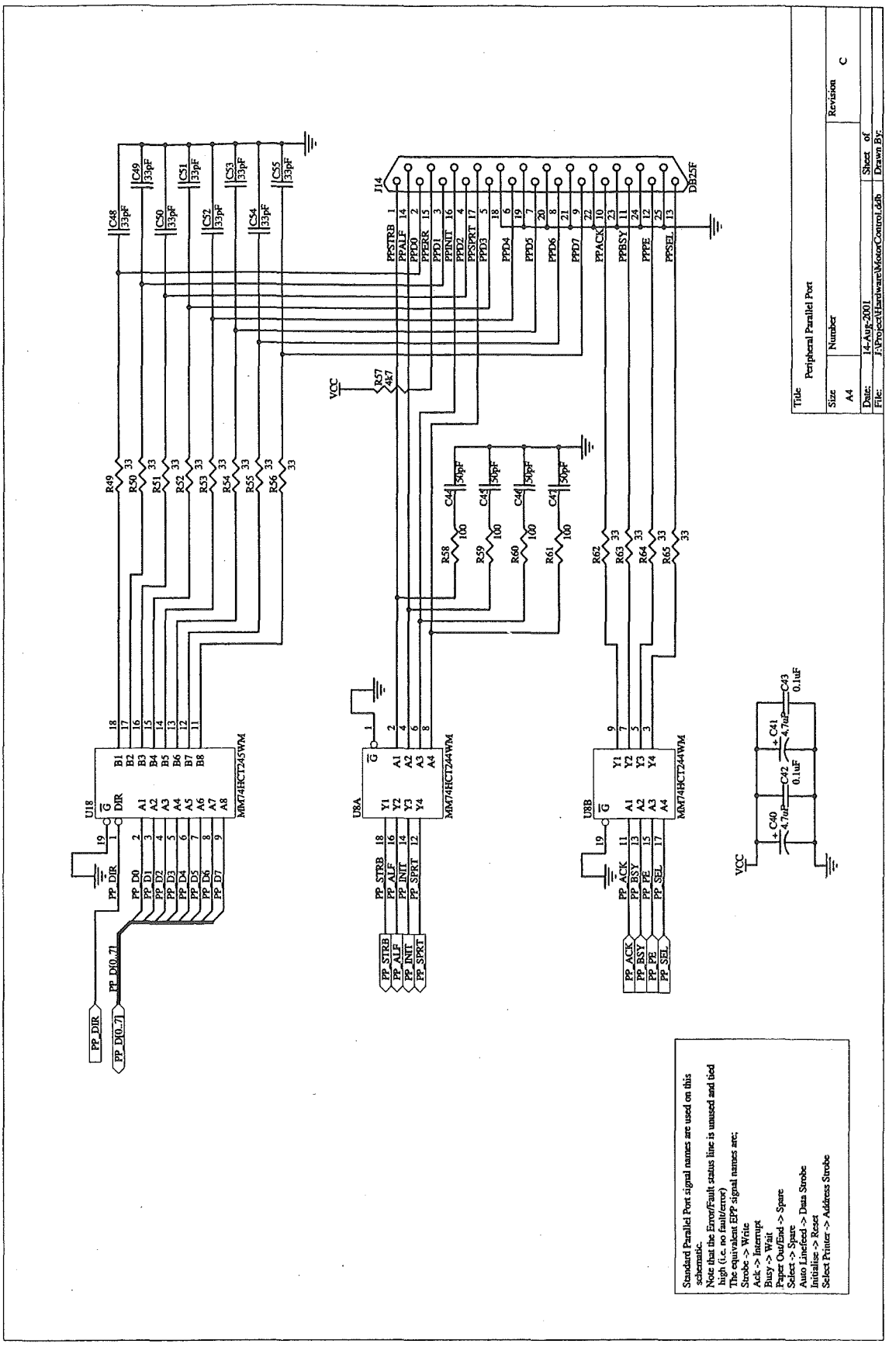


Figure B2.5: Analogue Expansion Board Schematic - Parallel Port Interface

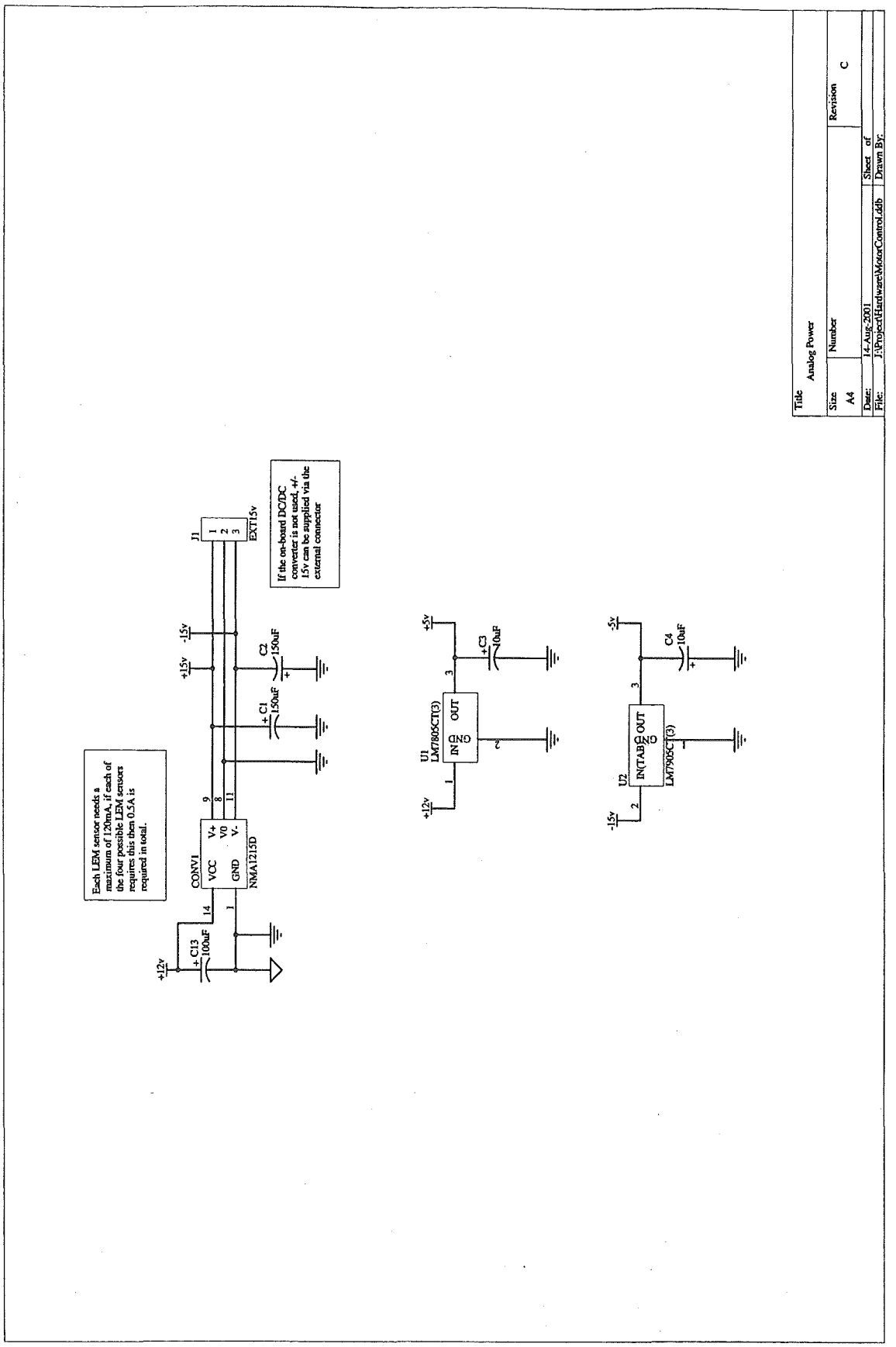


Figure B2.6: Analogue Expansion Board Schematic - Power Supplies

B-3. Analogue Signal Adapter Boards

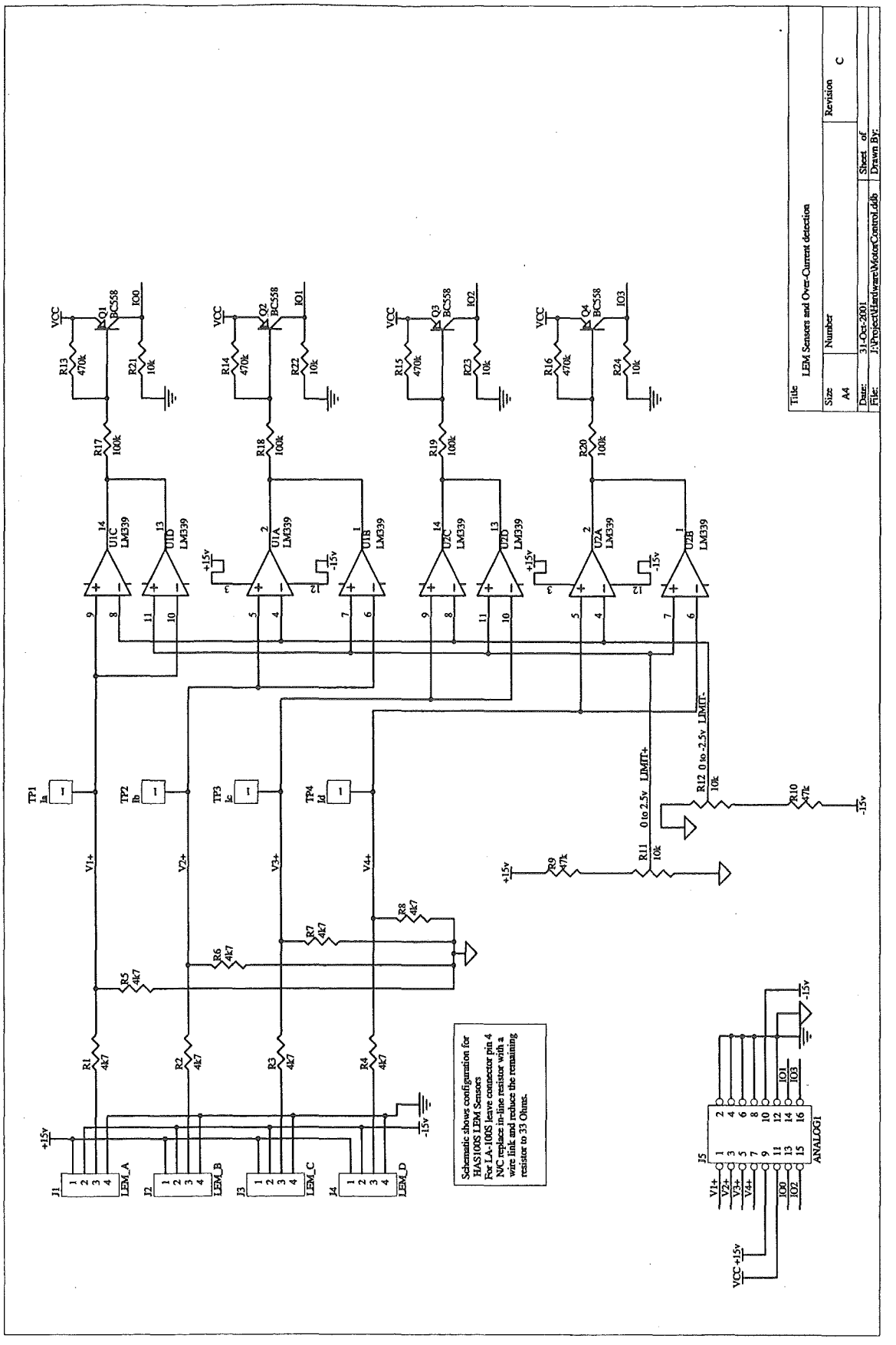


Figure B3.1: Analogue Signal Adapter Board Schematic - Current Measurement

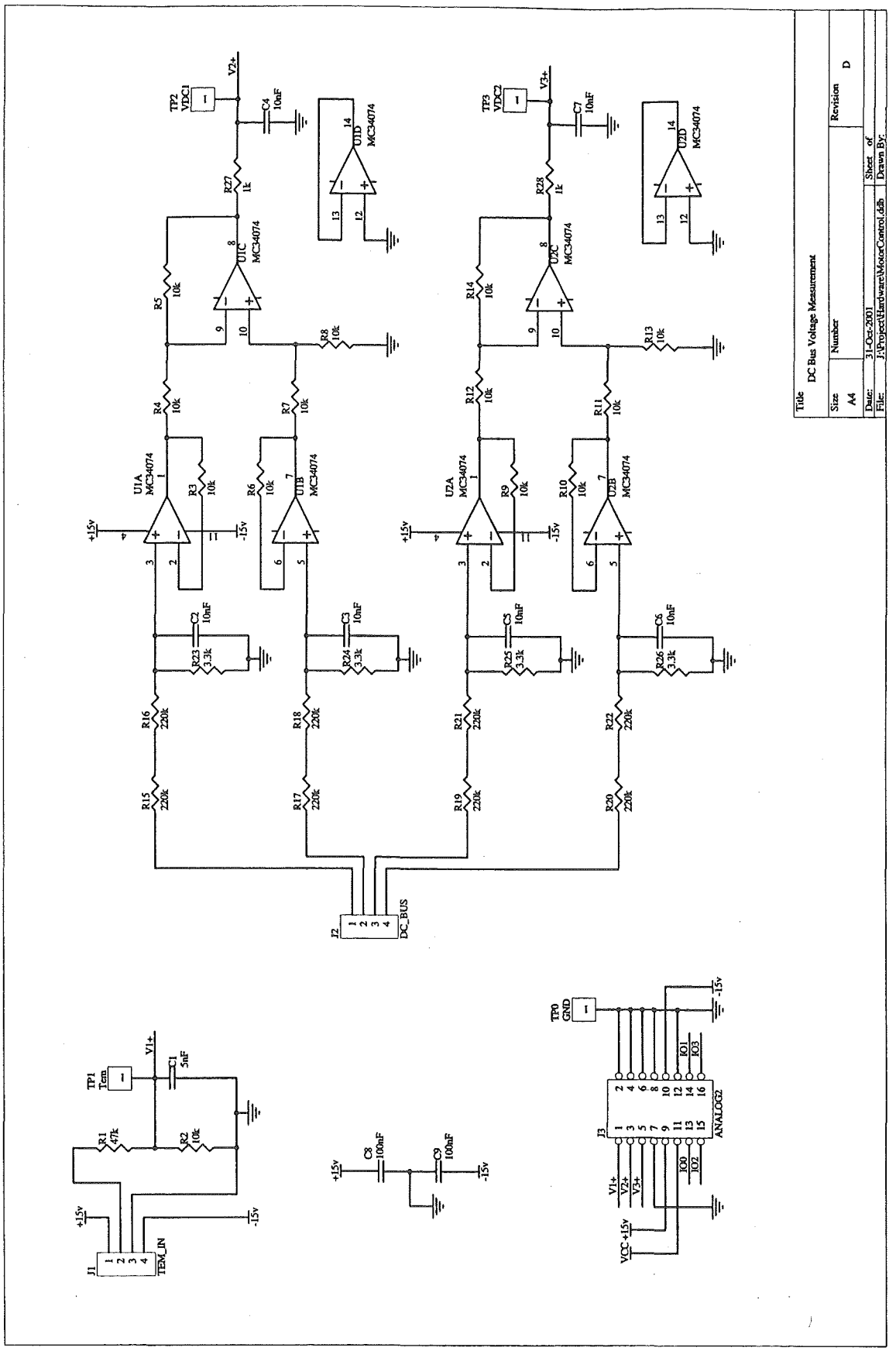


Figure B3.2: Analogue Signal Adapter Board Schematic - Voltage Measurement

Appendix C. FPGA Schematics

C-1. Processor Board FPGA Logic

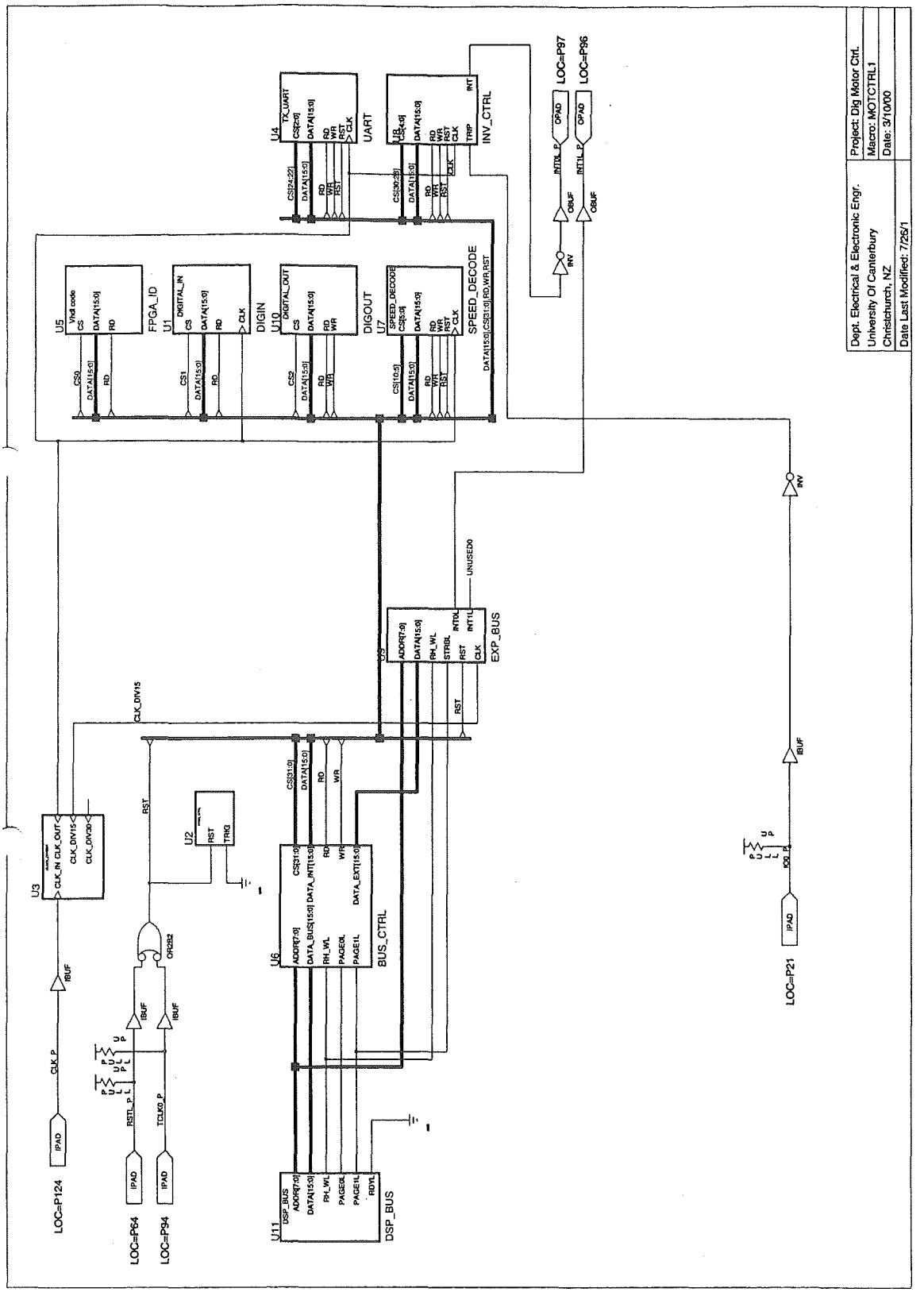


Figure C1.1: Processor Board FPGA - Top Level

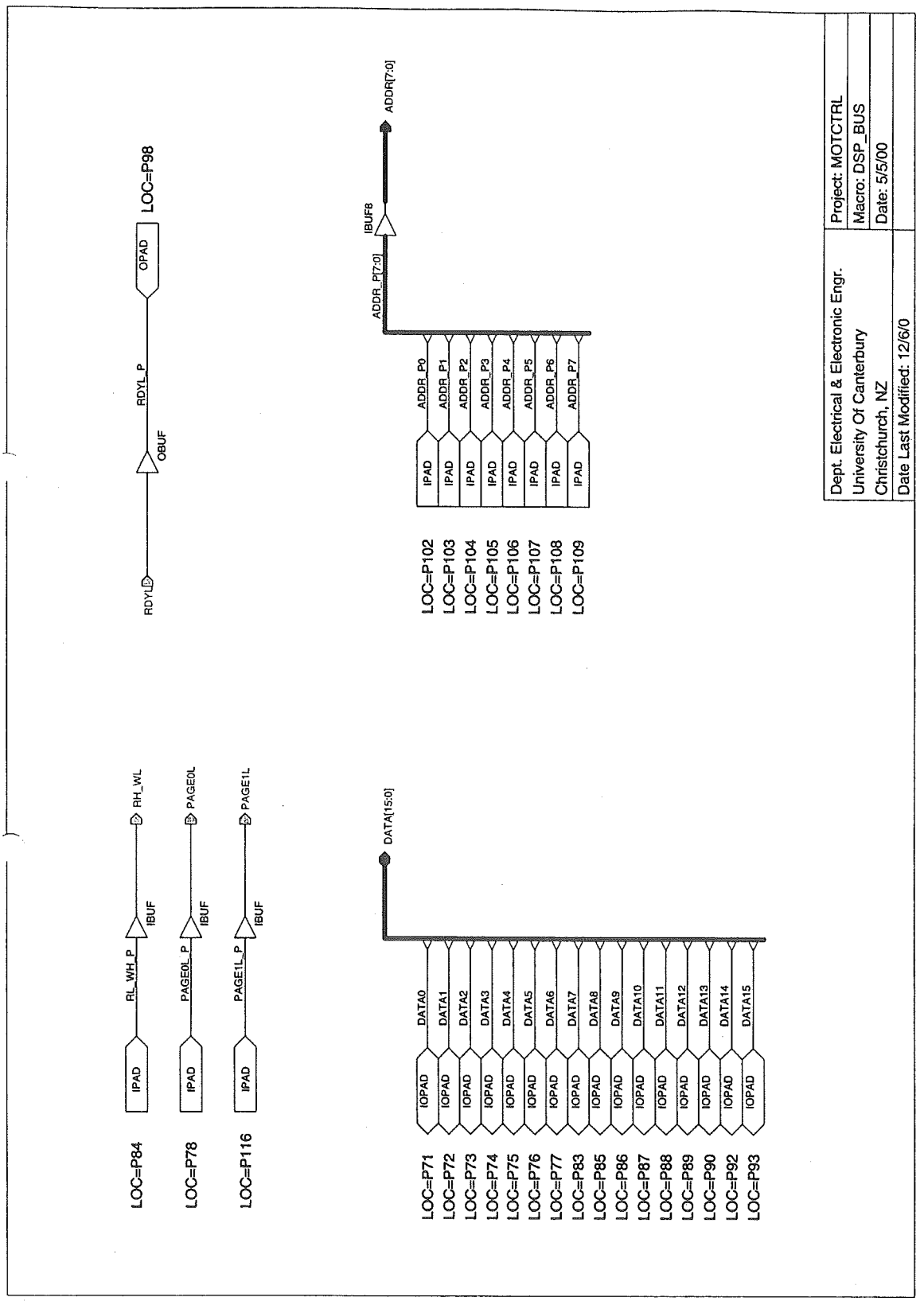


Figure C1.2: Processor Board FPGA - DSP Bus Interface

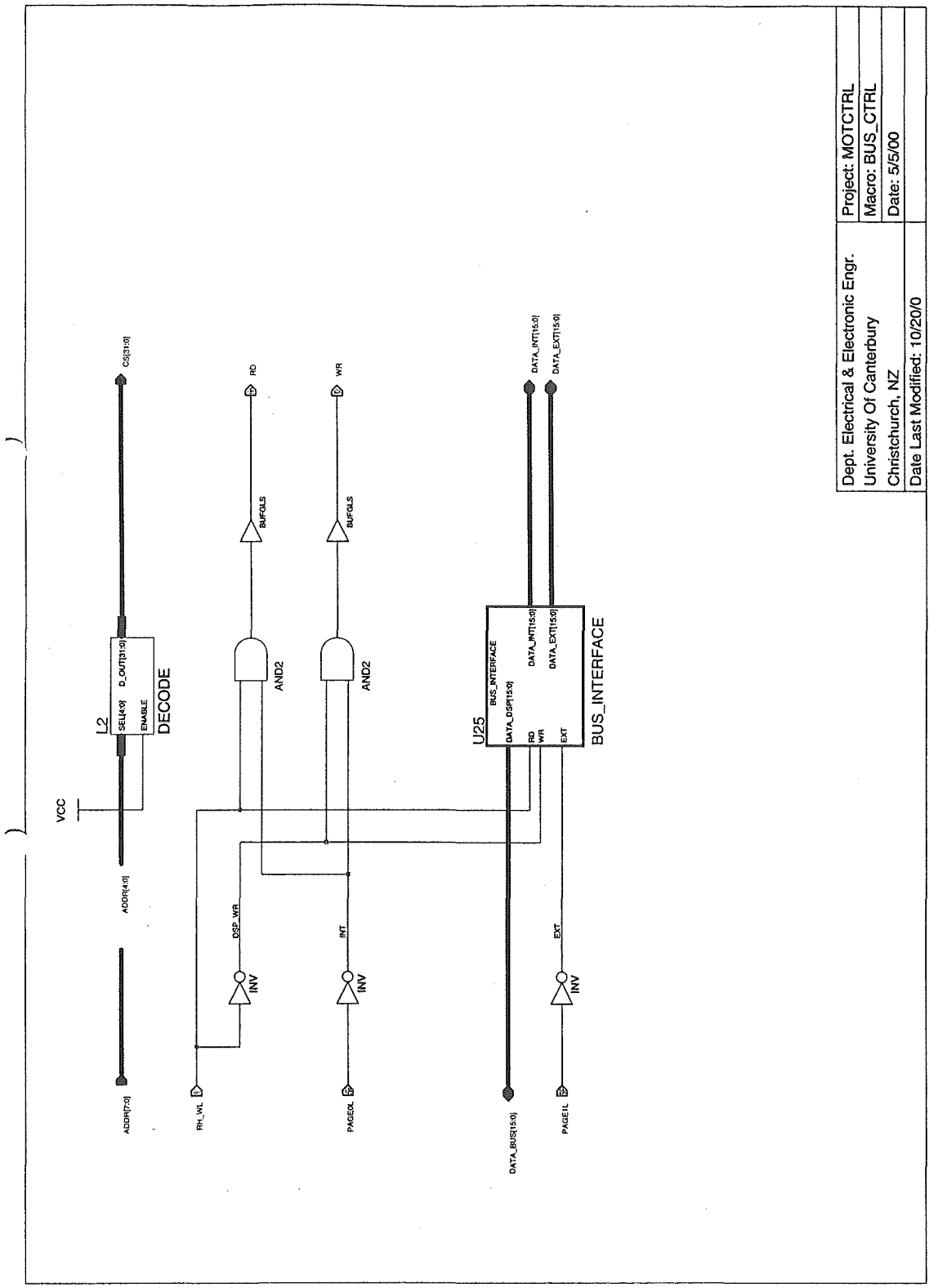


Figure C1.3: Processor Board FPGA - Bus Control

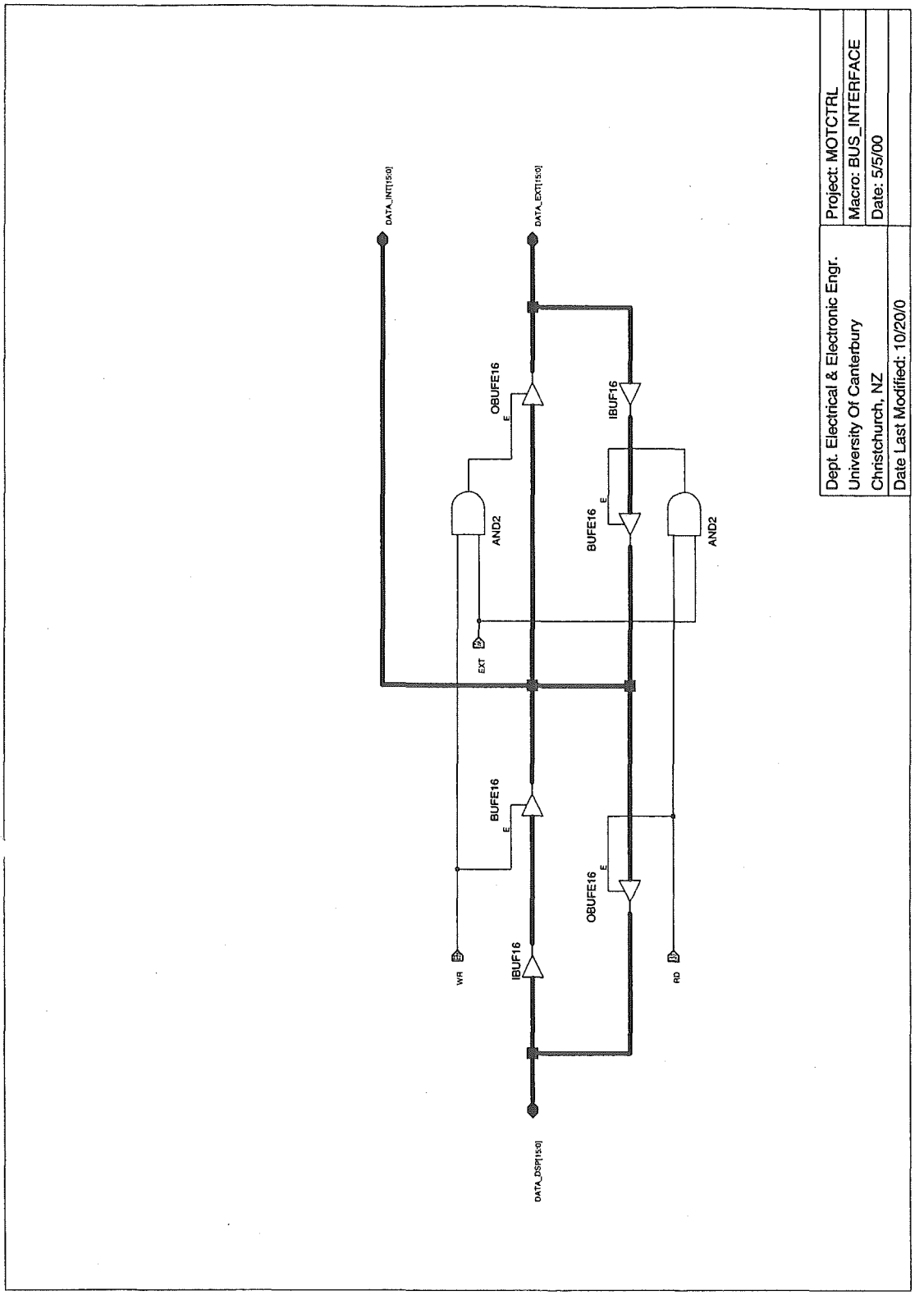


Figure C1.4: Processor Board FPGA - Bus Interface

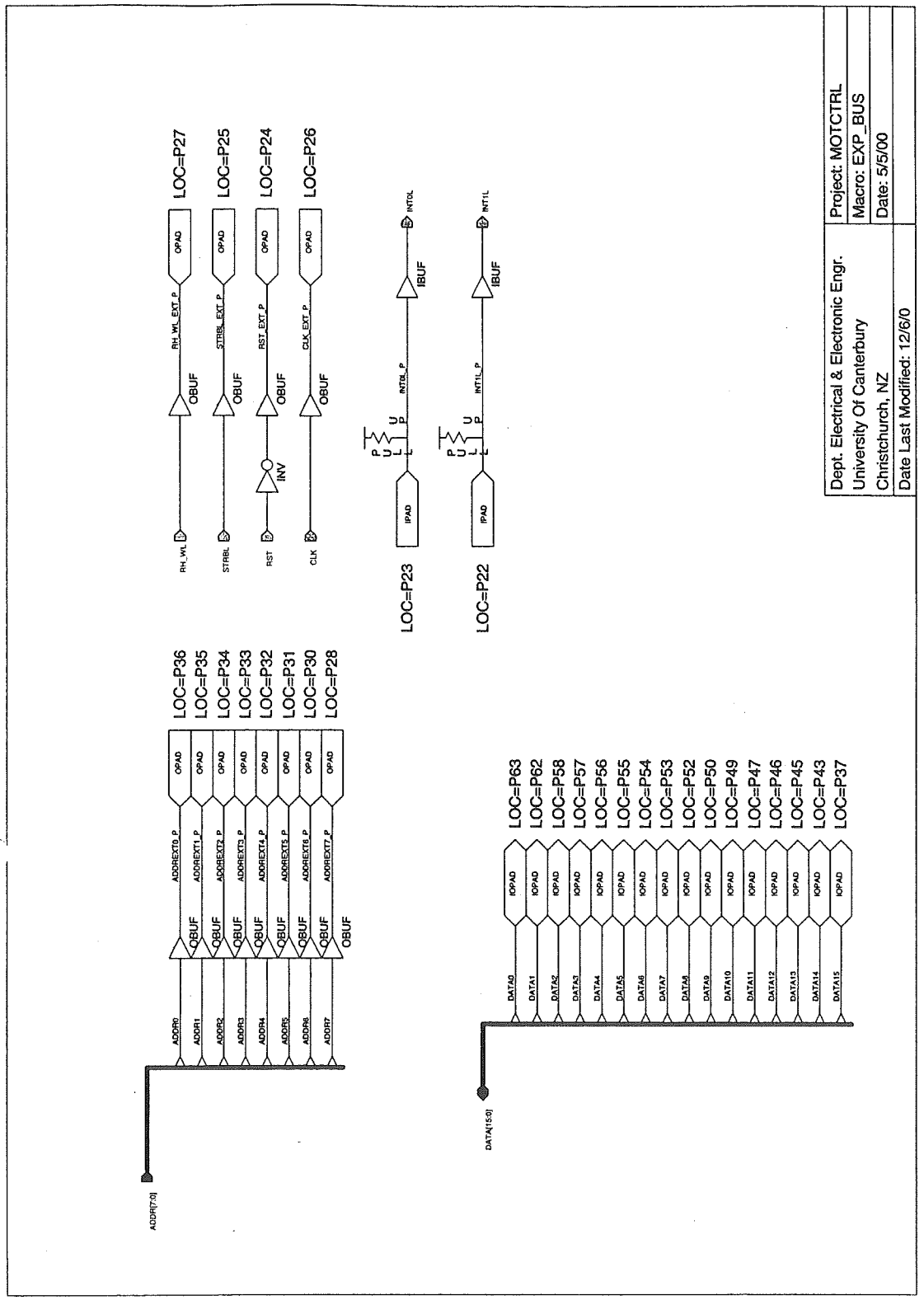


Figure C1.5: Processor Board FPGA - Expansion Bus

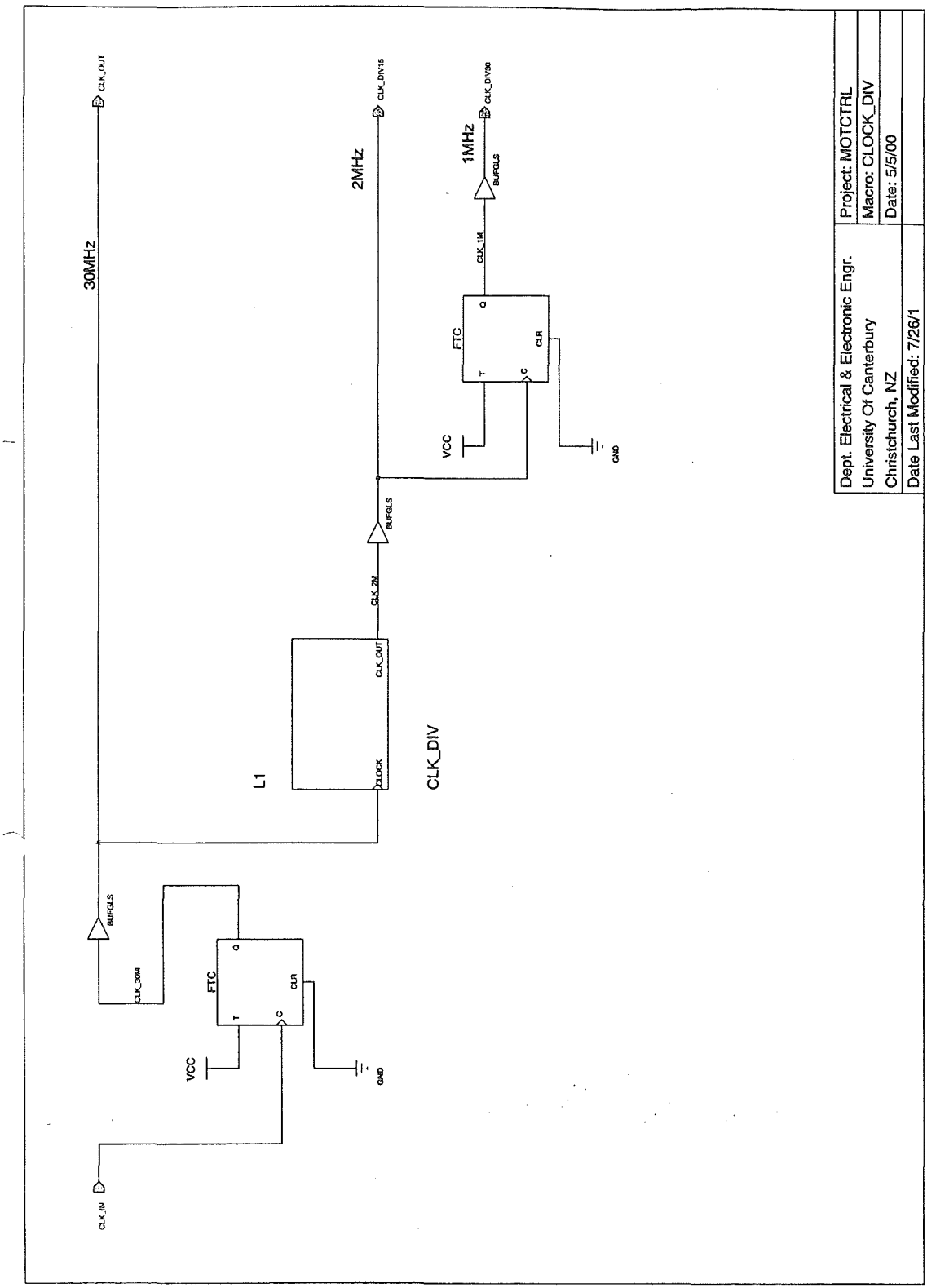


Figure C1.6: Processor Board FPGA - Clock Dividers

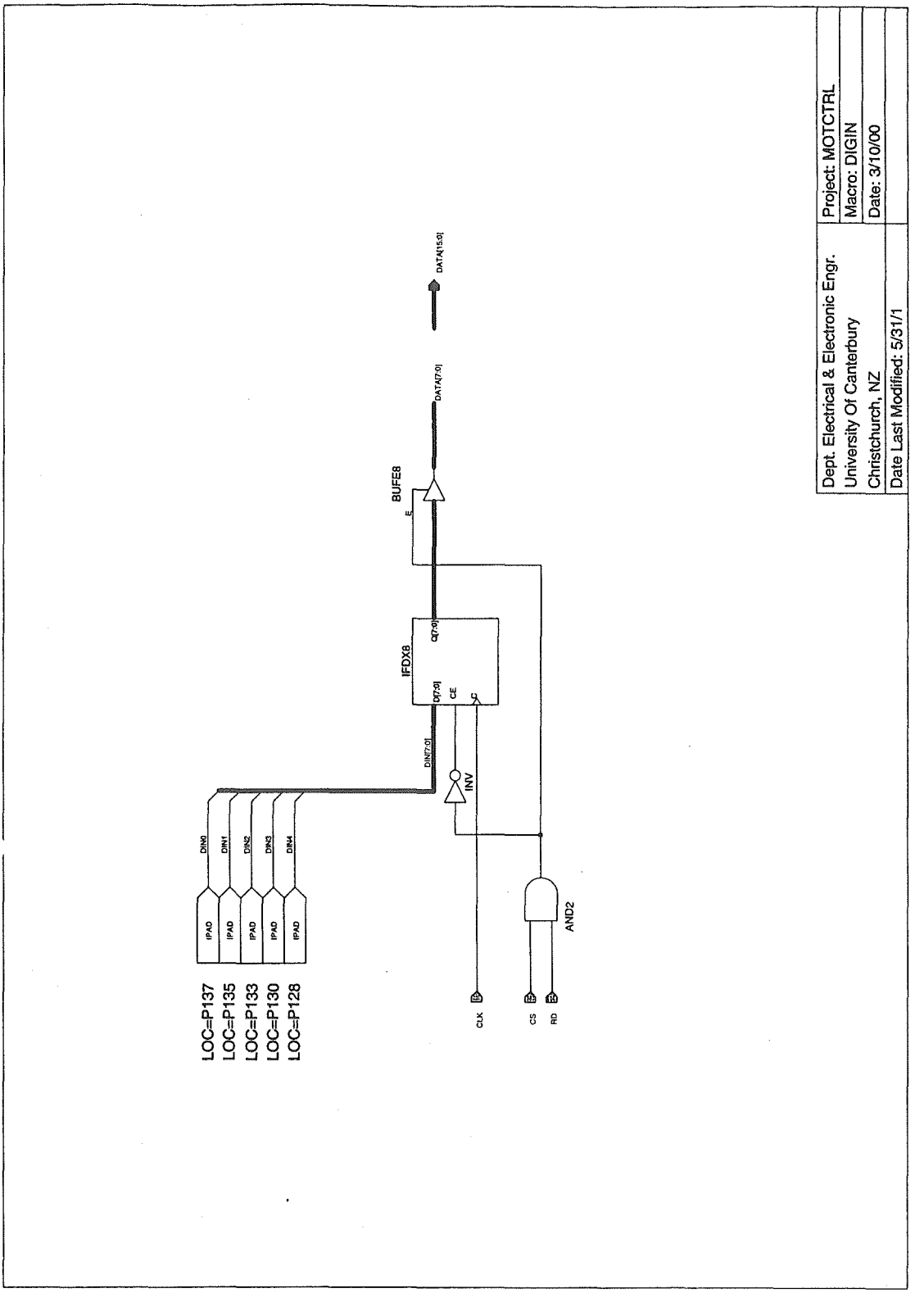


Figure C1.7: Processor Board FPGA - Digital Inputs

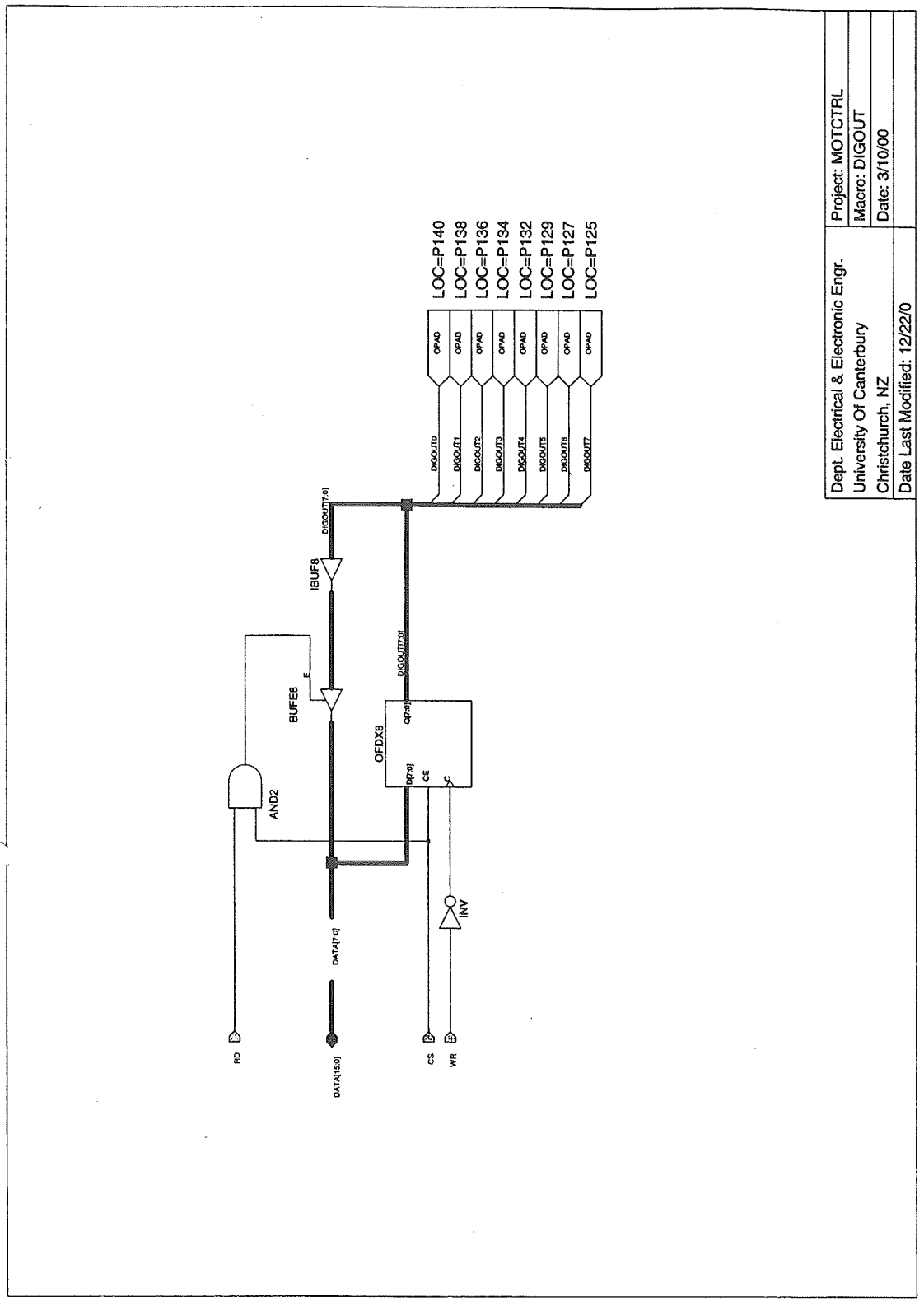


Figure C1.8: Processor Board FPGA - Digital Outputs

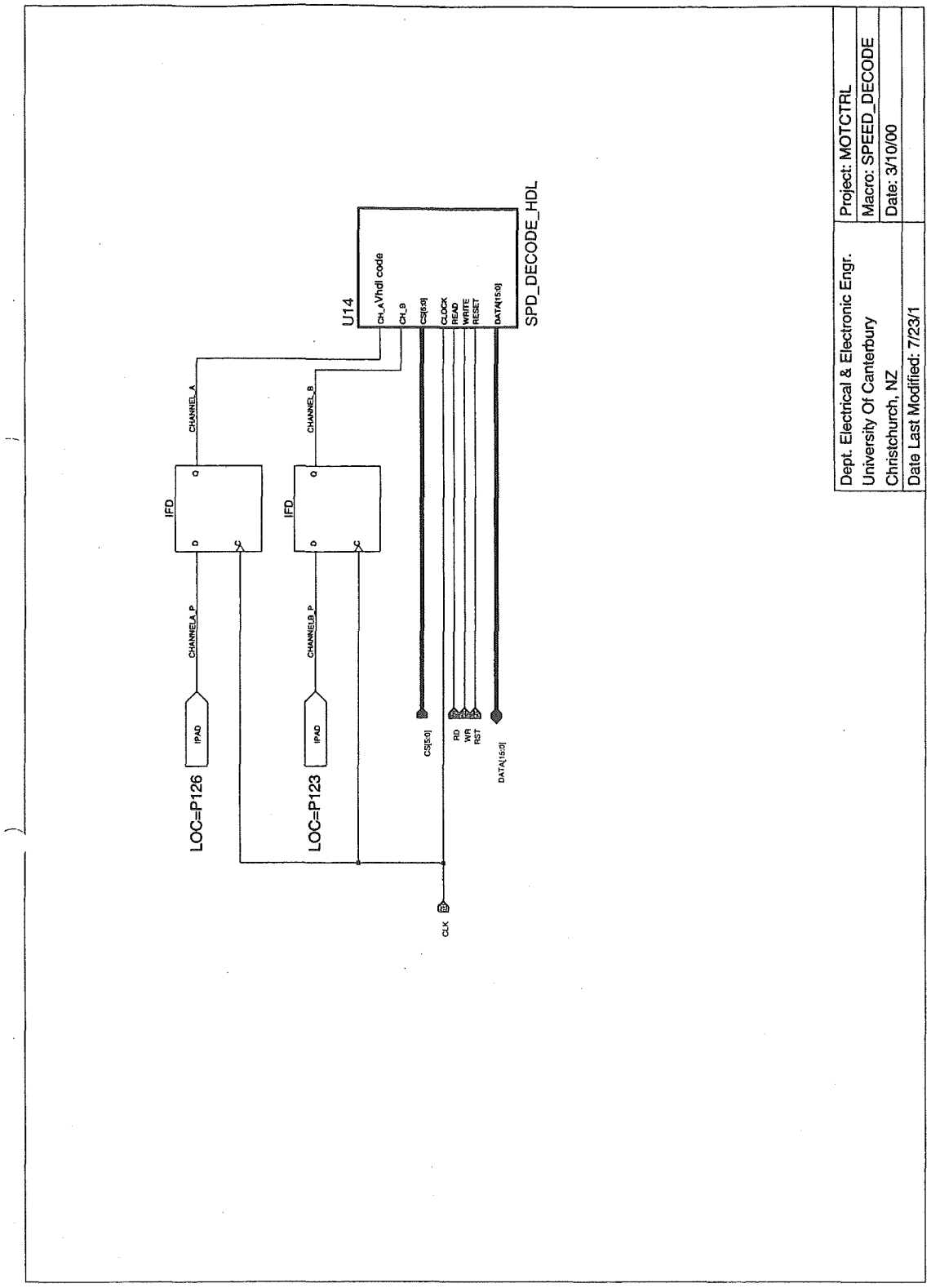


Figure C1.9: Processor Board FPGA - Speed Decoder

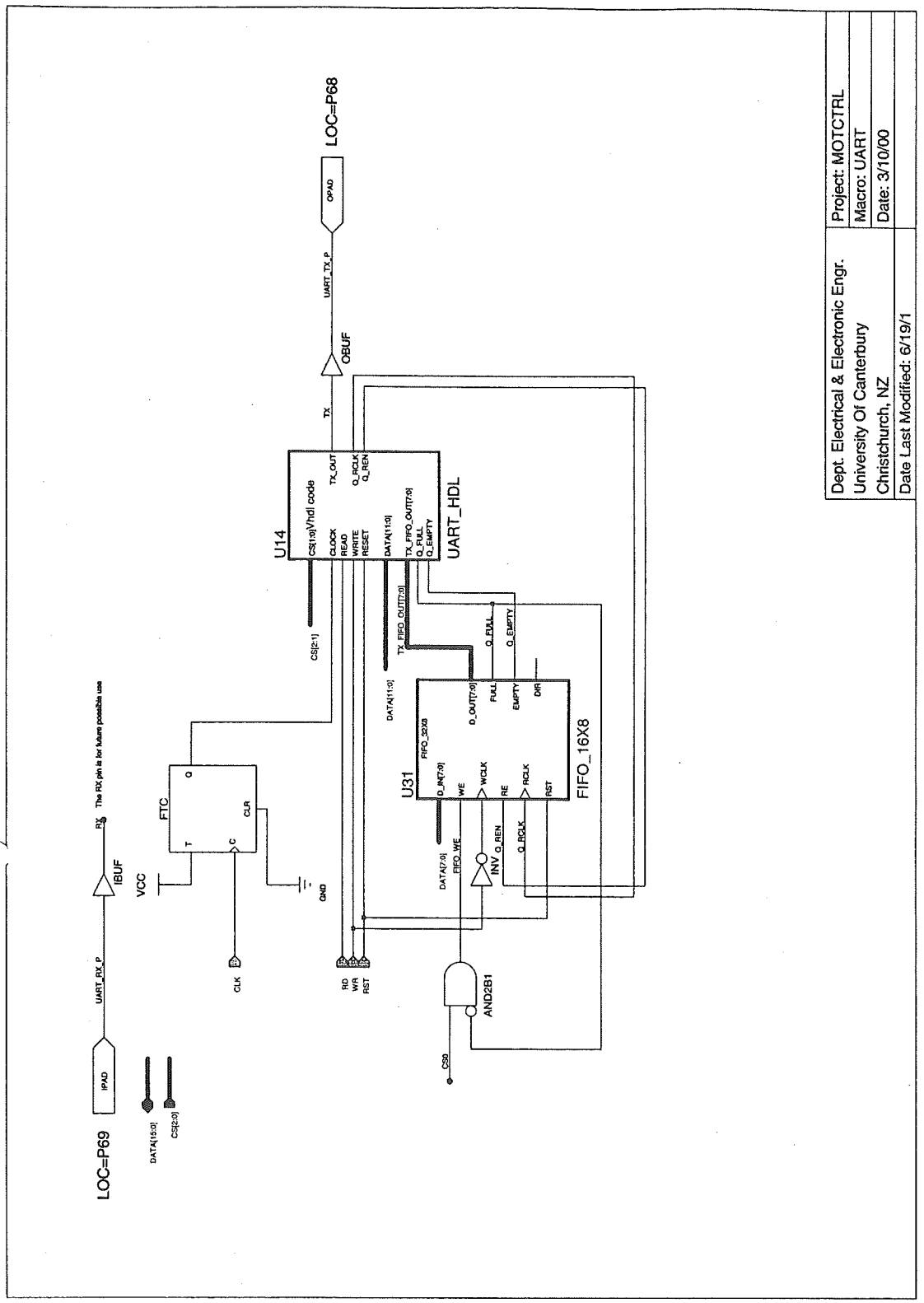


Figure C1.10: Processor Board FPGA - Serial Transmitter

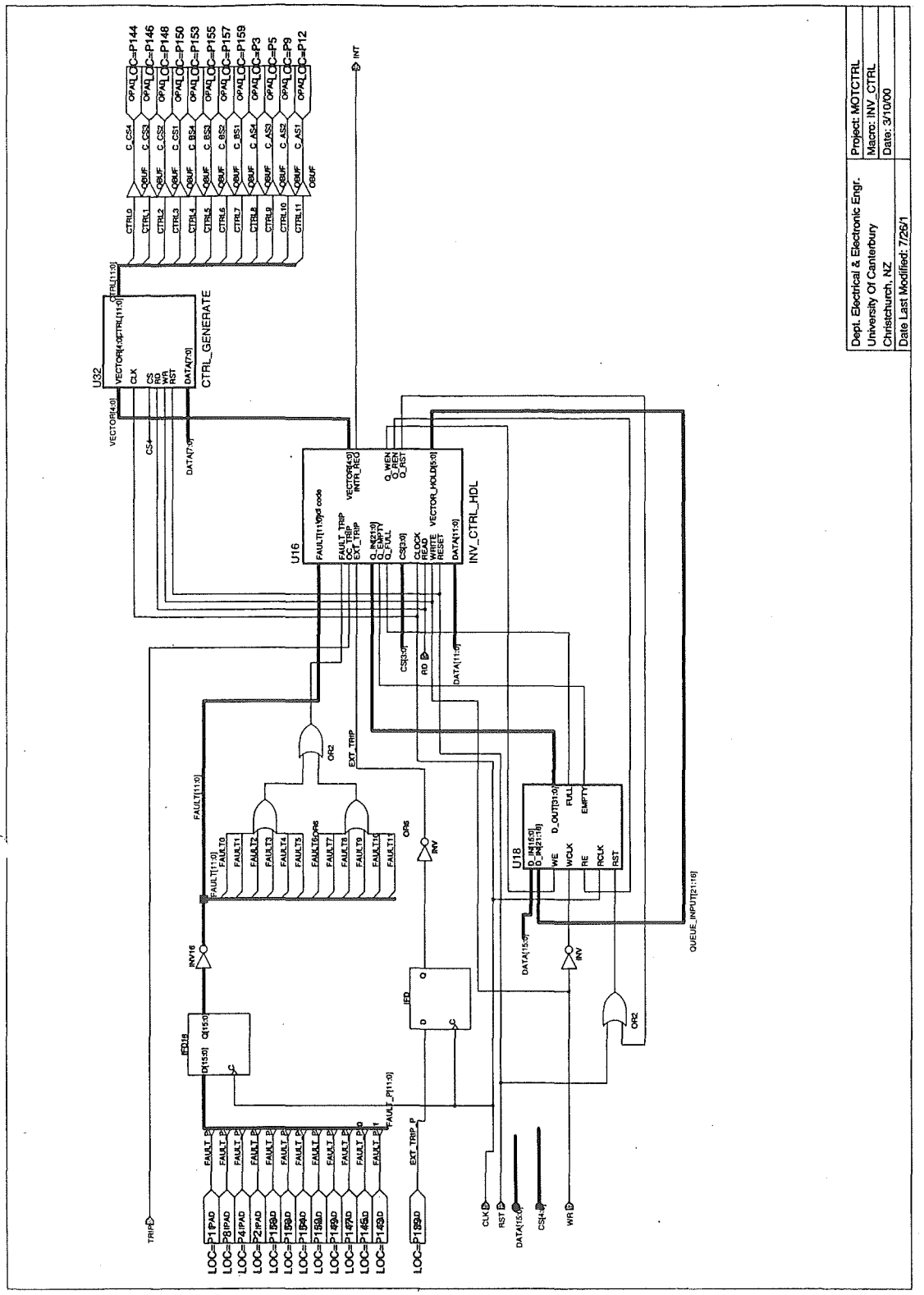


Figure C1.11: Processor Board FPGA - Inverter IGBT Timing Generation

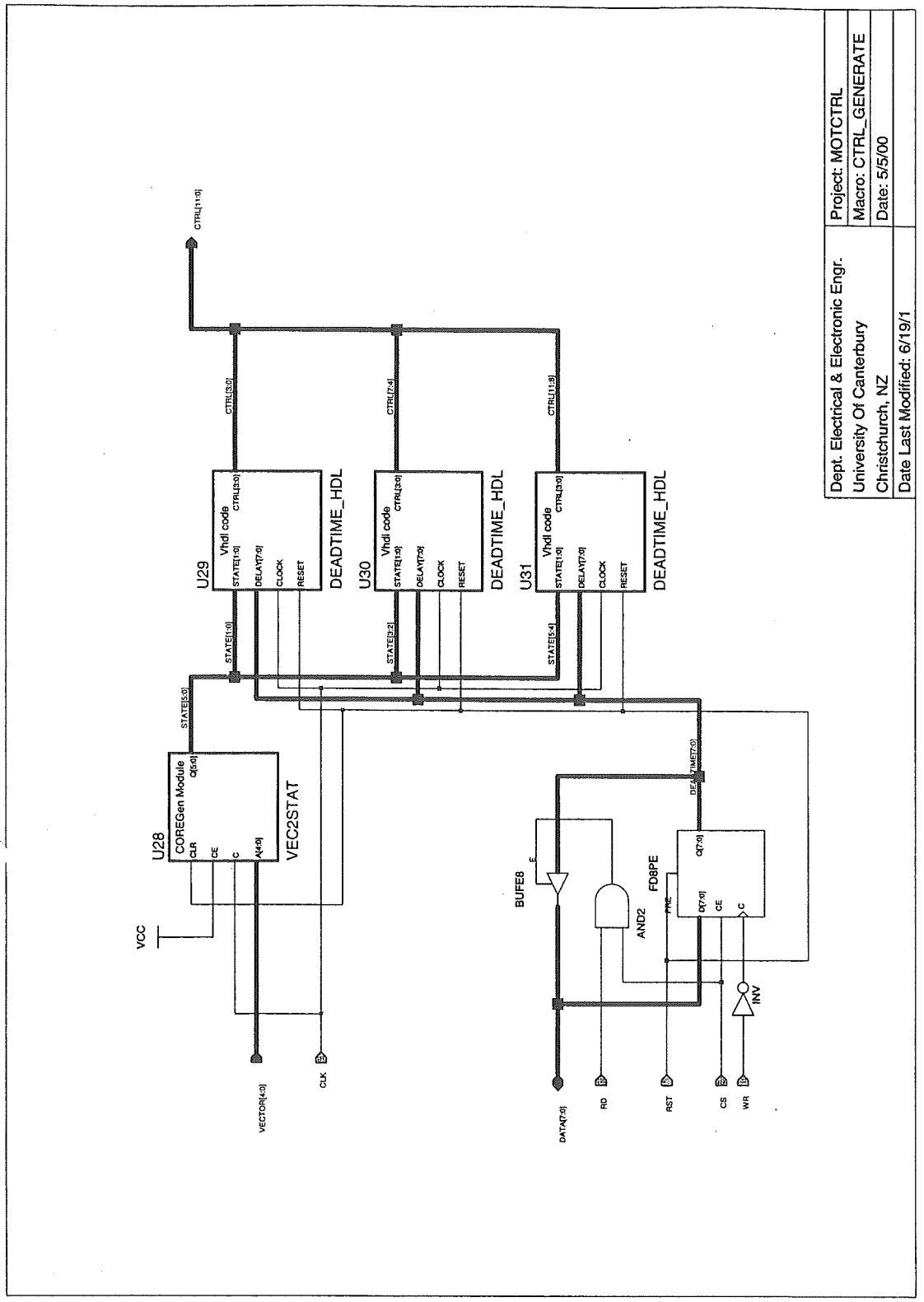


Figure C1.12: Processor Board FPGA - Timing Signal Generation

C-2. Analogue Expansion Board FPGA Logic

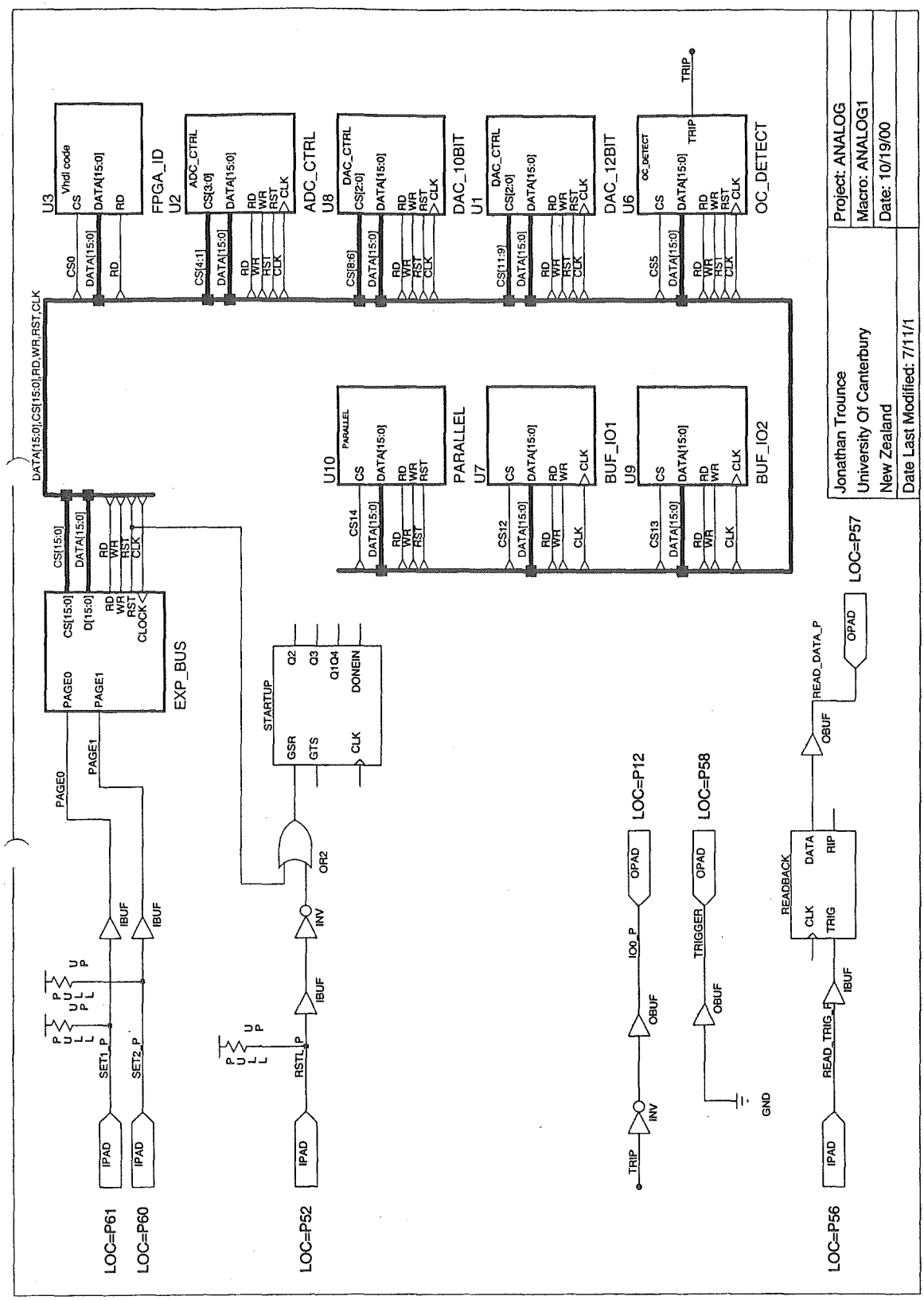


Figure C2.1: Analogue Expansion Board FPGA - Top Level

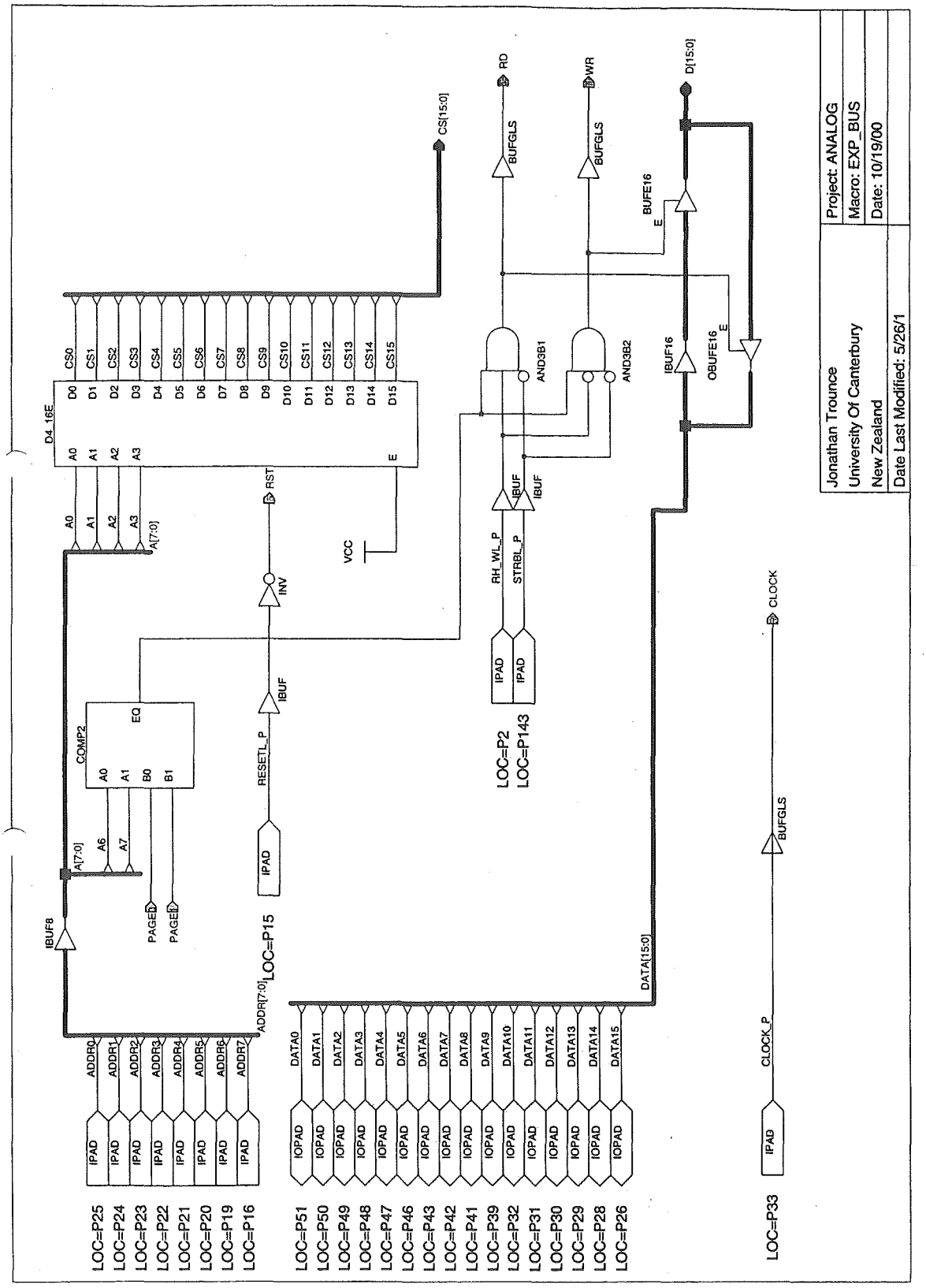


Figure C2.2: Analogue Expansion Board FPGA - Bus Interface

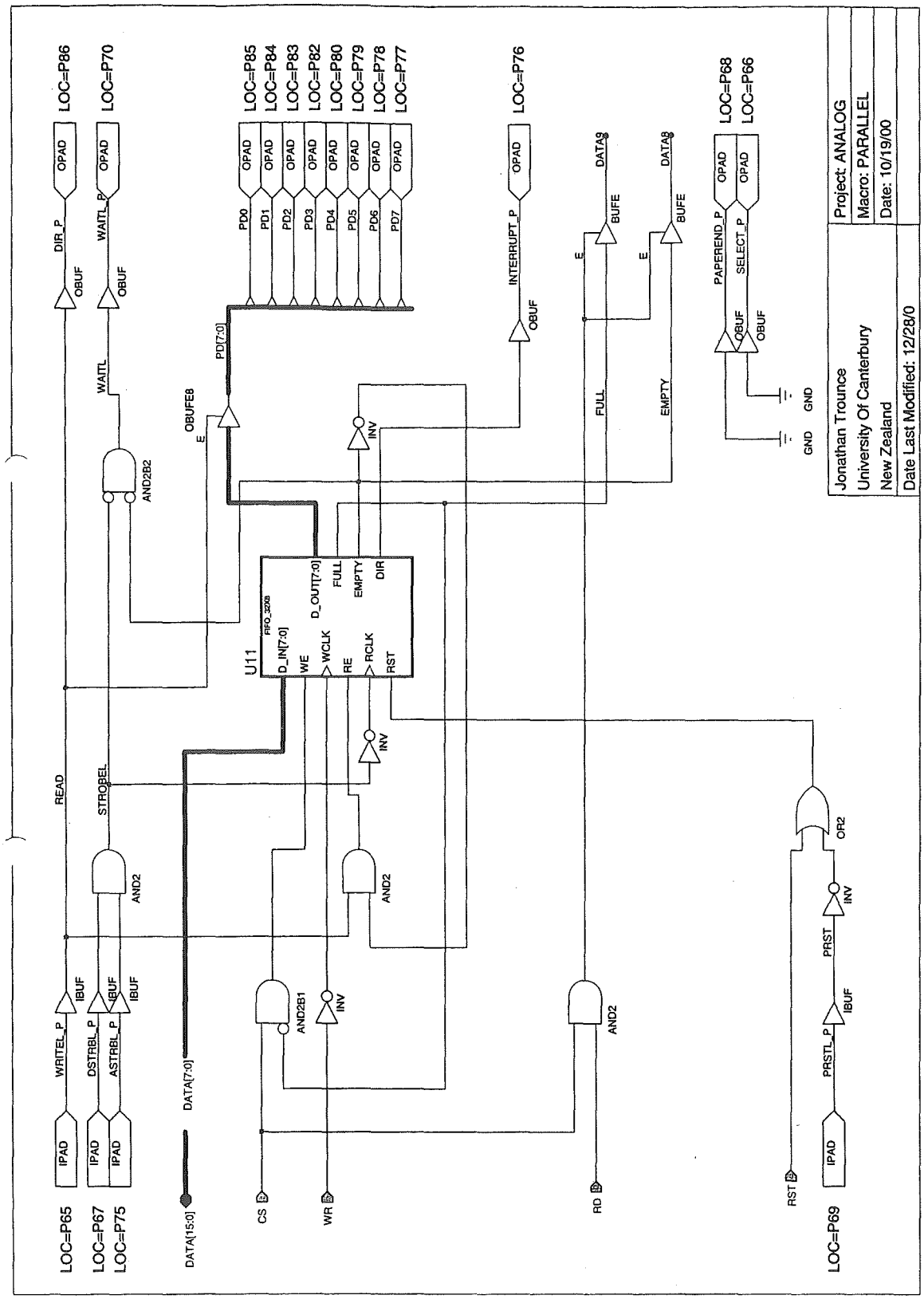


Figure C2.3: Analogue Expansion Board FPGA - Parallel Port Interface

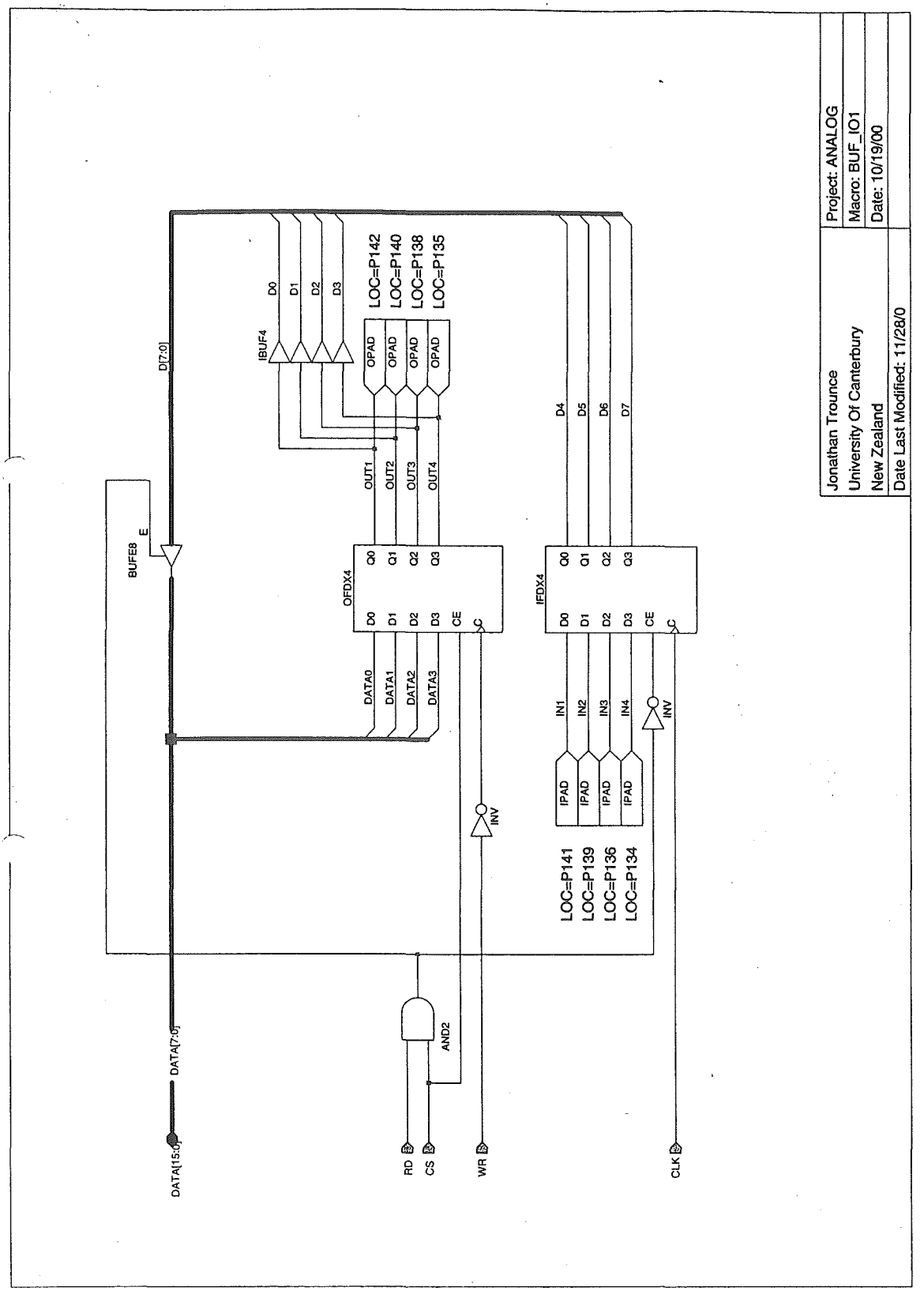


Figure C2.4: Analogue Expansion Board FPGA - IO Port 1

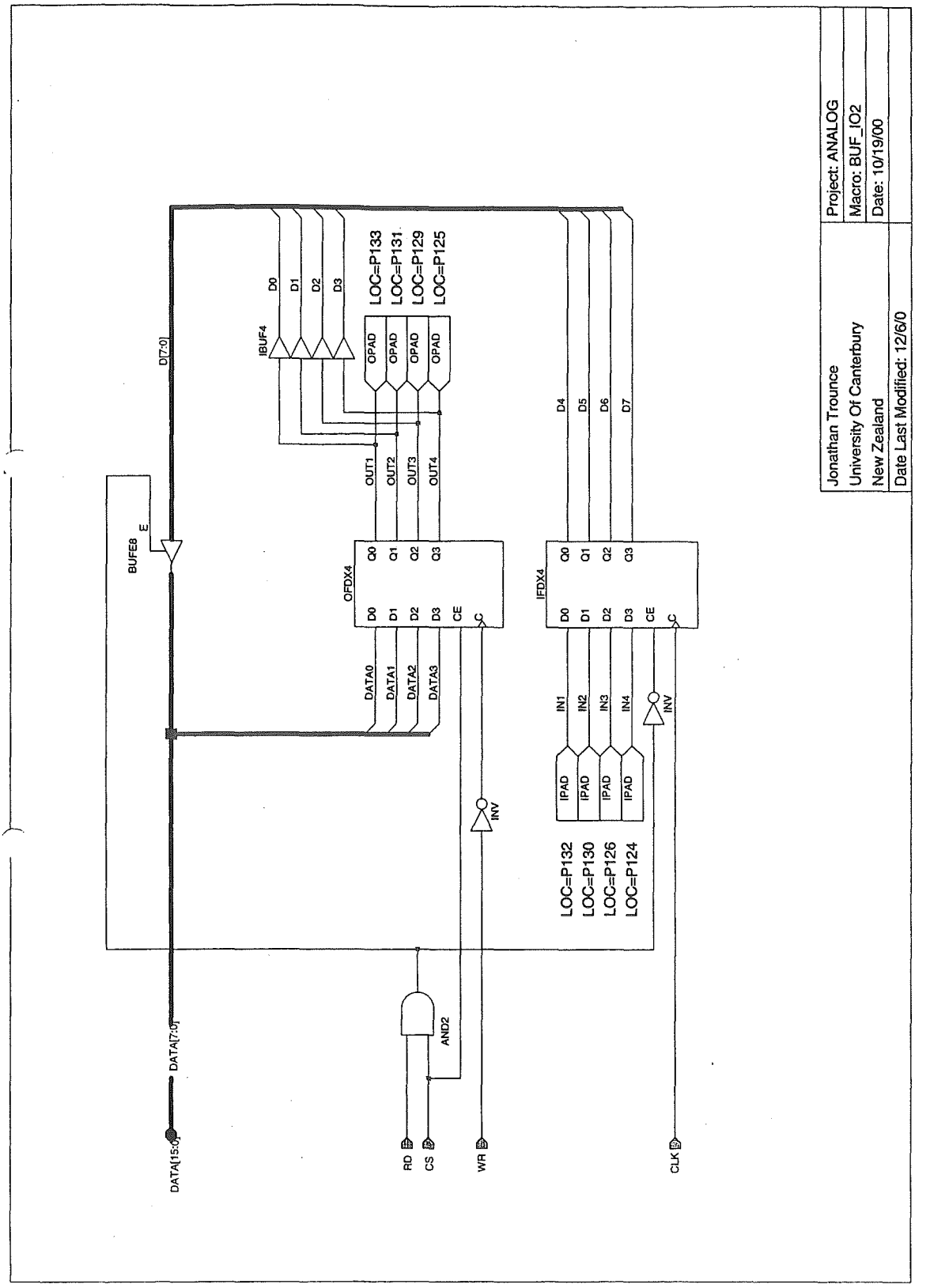


Figure C2.5: Analogue Expansion Board FPGA - IO Port 2

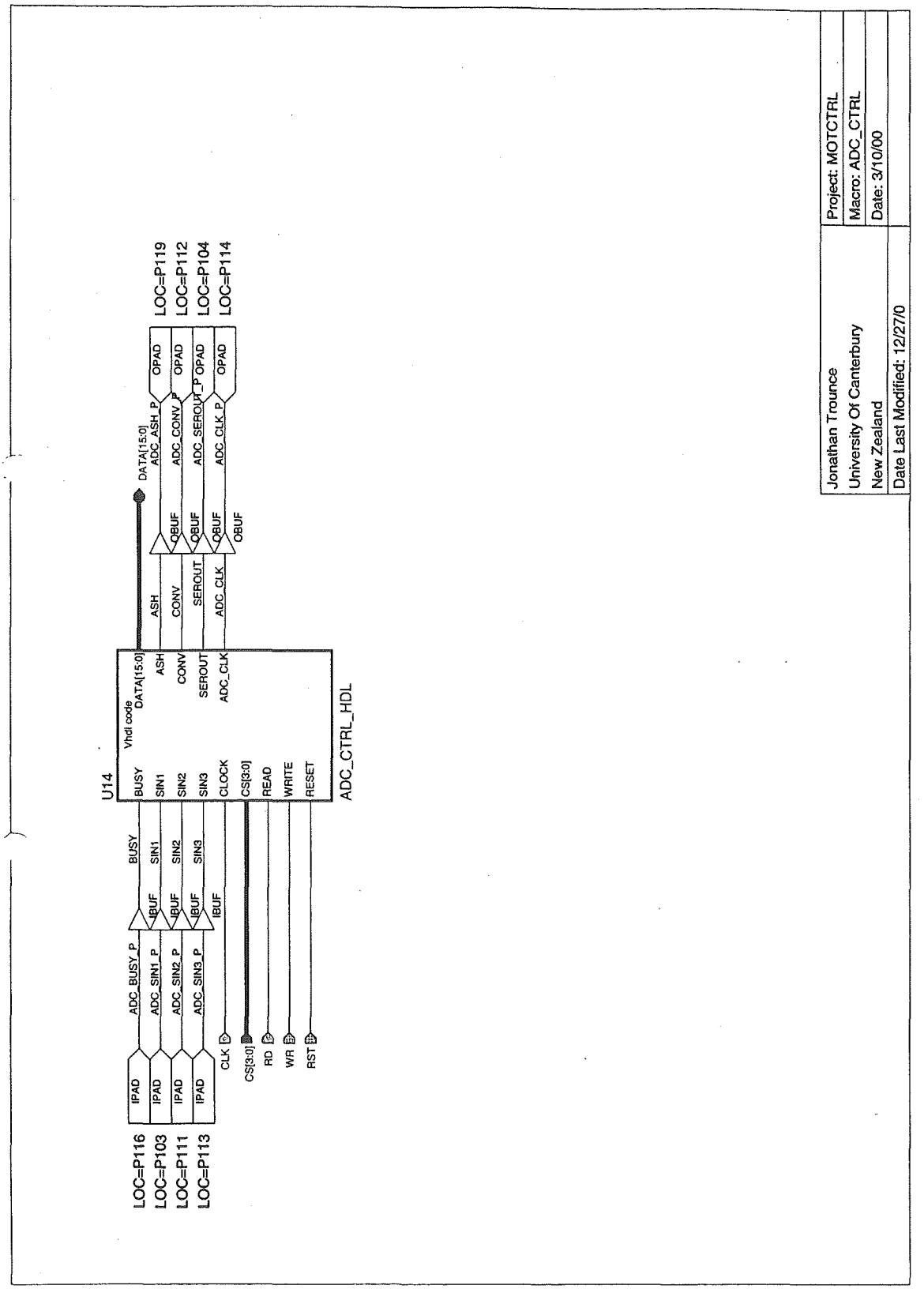


Figure C2.6: Analogue Expansion Board FPGA - ADC Control

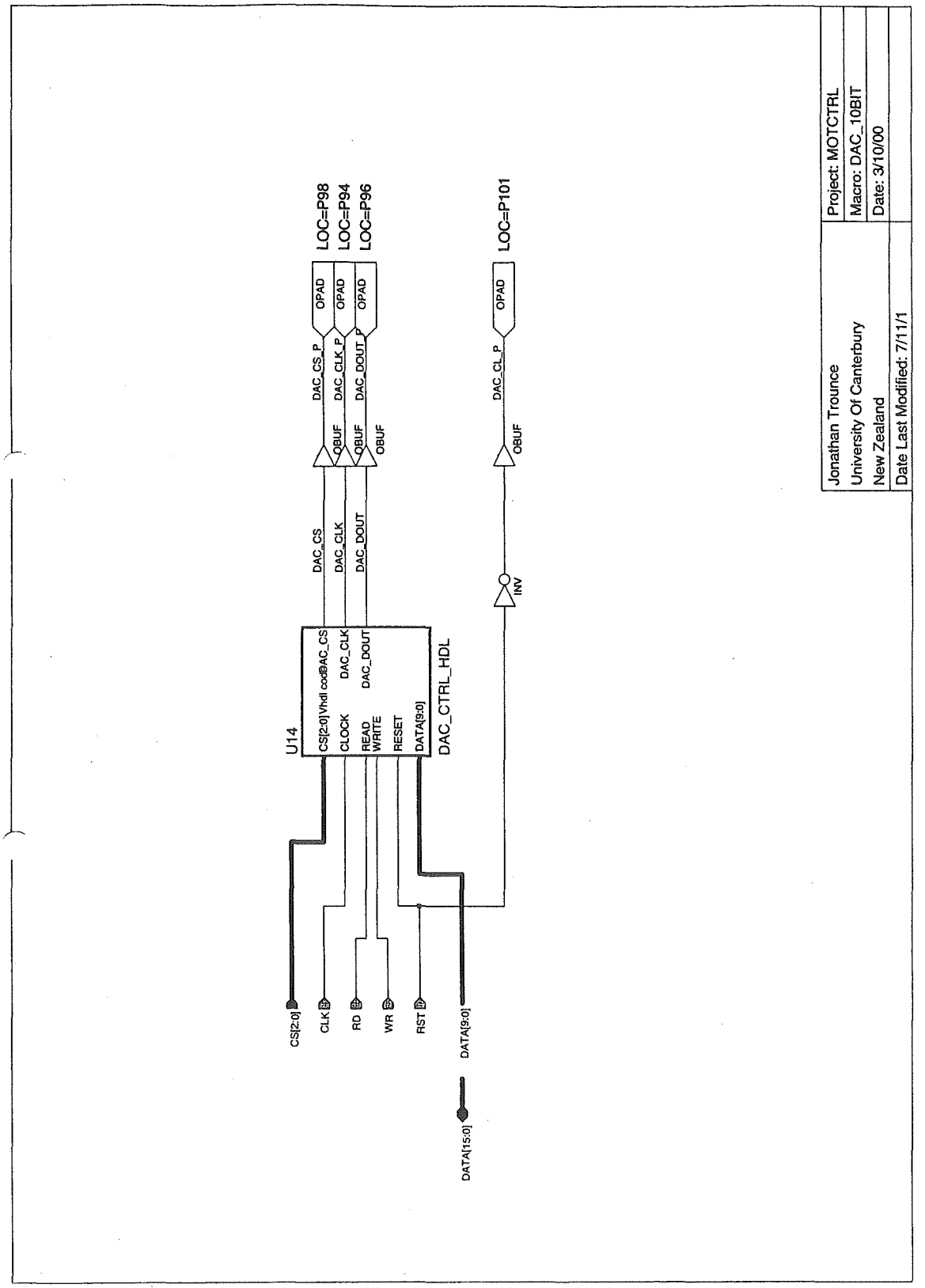


Figure C2.7: Analogue Expansion Board FPGA - 10-bit DAC Control

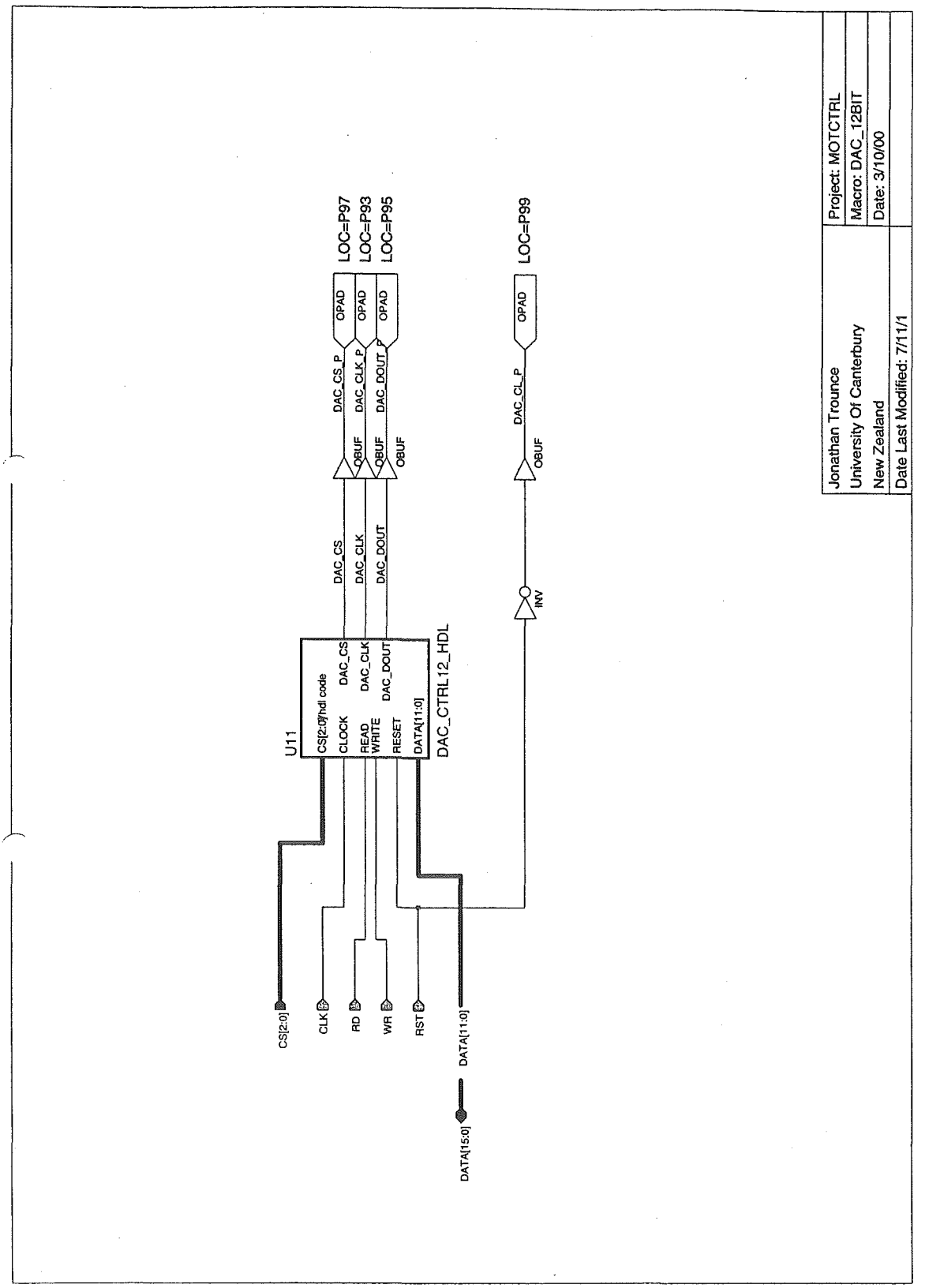


Figure C2.8: Analogue Expansion Board FPGA - 12-bit DAC Control

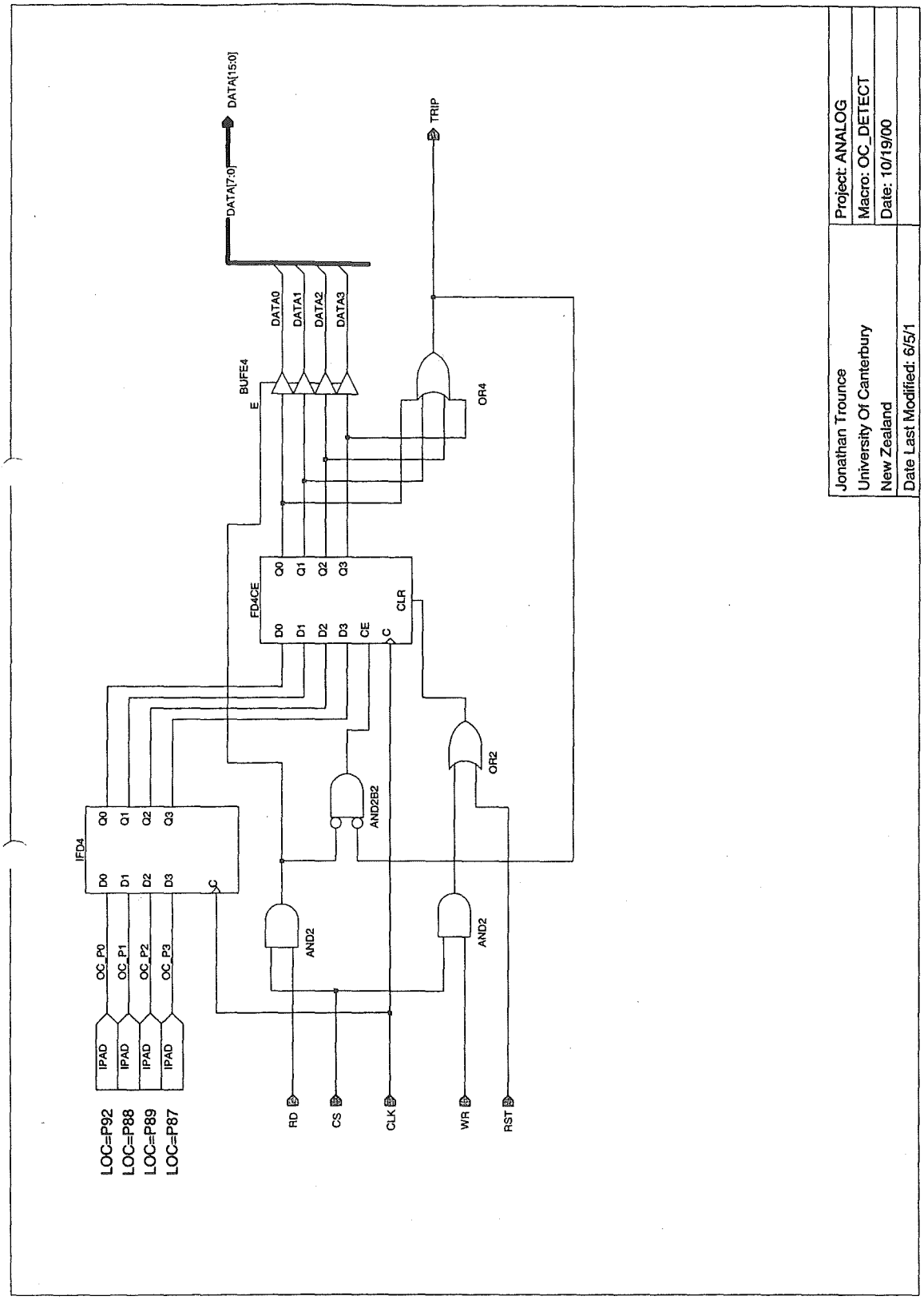


Figure C2.9: Analogue Expansion Board FPGA - Overcurrent Detection

TMS320VC33
DIGITAL SIGNAL PROCESSOR

SPRS087B – FEBRUARY 1999 – REVISED JULY 2000

- High-Performance Floating-Point Digital Signal Processor (DSP):
  - TMS320VC33-150
    - 13-ns Instruction Cycle Time
    - 150 Million Floating-Point Operations Per Second (MFLOPS)
    - 75 Million Instructions Per Second (MIPS)
  - TMS320VC33-120
    - 17-ns Instruction Cycle Time
    - 120 MFLOPS
    - 60 MIPS
- 34K × 32-Bit (1.1-Mbit) On-Chip Words of Dual-Access Static Random-Access Memory (SRAM) Configured in 2 × 16K plus 2 × 1K Blocks to Improve Internal Performance
- x5 Phase-Locked Loop (PLL) Clock Generator
- Very Low Power: < 200 mW @ 150 MFLOPS
- 32-Bit High-Performance CPU
- 16-/32-Bit Integer and 32-/40-Bit Floating-Point Operations
- Four Internally Decoded Page Strobes to Simplify Interface to I/O and Memory Devices
- Boot-Program Loader
- EDGEMODE Selectable External Interrupts
- 32-Bit Instruction Word, 24-Bit Addresses
- Eight Extended-Precision Registers
- On-Chip Memory-Mapped Peripherals:
  - One Serial Port
  - Two 32-Bit Timers
  - Direct Memory Access (DMA) Coprocessor for Concurrent I/O and CPU Operation
- Fabricated Using the 0.18-µm (l_{eff}-Effective Gate Length) Timeline™ Technology by Texas Instruments (TI)
- 144-Pin Low-Profile Quad Flatpack (LQFP) (PGE Suffix)
- Two Address Generators With Eight Auxiliary Registers and Two Auxiliary Register Arithmetic Units (ARAUs)
- Two Low-Power Modes
- Two- and Three-Operand Instructions
- Parallel Arithmetic/Logic Unit (ALU) and Multiplier Execution in a Single Cycle
- Block-Repeat Capability
- Zero-Overhead Loops With Single-Cycle Branches
- Conditional Calls and Returns
- Interlocked Instructions for Multiprocessing Support
- Bus-Control Registers Configure Strobe-Control Wait-State Generation
- 1.8-V (Core) and 3.3-V (I/O) Supply Voltages
- On-Chip Scan-Based Emulation Logic, IEEE Std 1149.1† (JTAG)

description

The TMS320VC33 DSP is a 32-bit, floating-point processor manufactured in 0.18-µm four-level-metal CMOS (Timeline) technology. The TMS320VC33 is part of the TMS320C3x generation of DSPs from Texas Instruments.

The TMS320C3x's internal busing and special digital-signal-processing instruction set have the speed and flexibility to execute up to 150 million floating-point operations per second (MFLOPS). The TMS320VC33 optimizes speed by implementing functions in hardware that other processors implement through software or microcode. This hardware-intensive approach provides performance previously unavailable on a single chip.

The TMS320VC33 can perform parallel multiply and ALU operations on integer or floating-point data in a single cycle. Each processor also possesses a general-purpose register file, a program cache, dedicated ARAUs, internal dual-access memories, one DMA channel supporting concurrent I/O, and a short machine-cycle time. High performance and ease of use are the results of these features.

Figure D.1: TMS320VC33 Datasheet - Page 1 of 2

TMS320VC33
DIGITAL SIGNAL PROCESSOR

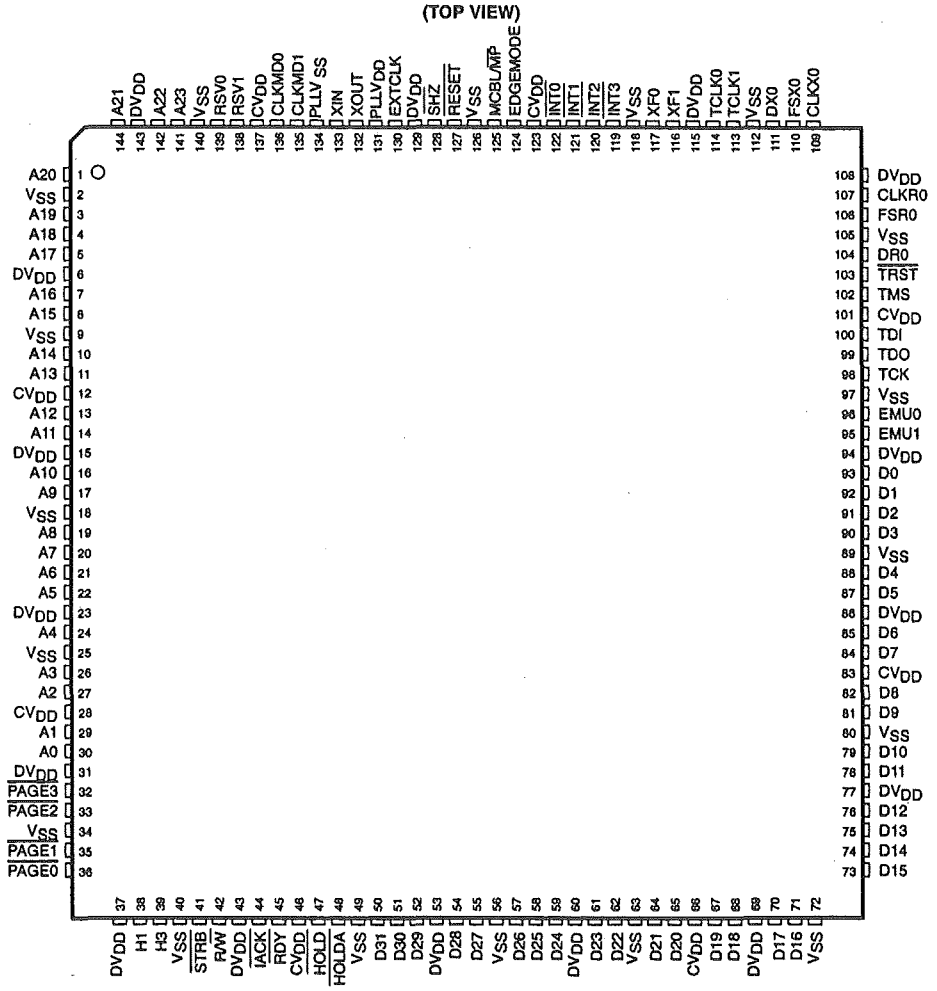
SPRS087B – FEBRUARY 1999 – REVISED JULY 2000

description (continued)

General-purpose applications are greatly enhanced by the large address space, multiprocessor interface, internally and externally generated wait states, one external interface port, two timers, one serial port, and multiple-interrupt structure. The TMS320C3x supports a wide variety of system applications from host processor to dedicated coprocessor. High-level-language support is easily implemented through a register-based architecture, large address space, powerful addressing modes, flexible instruction set, and well-supported floating-point arithmetic.

pinout

PGE PACKAGE†‡
(TOP VIEW)



† DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU. V_{SS} is the ground for both the I/O pins and the core CPU.
‡ $PLLV_{DD}$ and $PLLV_{SS}$ are isolated PLL supply pins that should be externally connected to CV_{DD} and V_{SS}, respectively.

The TMS320VC33 device is packaged in 144-pin low-profile quad flatpacks (PGE Suffix).

Figure D.2: TMS320VC33 Datasheet - Page 2 of 2

XILINX®

XC4000XLA/XV Field Programmable Gate Arrays

DS015 (v1.3) October 18, 1999

Product Specification

XC4000XLA/XV Family Features

Note: XC4000XLA devices are improved versions of XC4000XL devices. The XC4000XV devices have the same features as XLA devices, incorporate additional interconnect resources and extend gate capacity to 500,000 system gates. The XC4000XV devices require a separate 2.5V power supply for internal logic but maintain 5V I/O compatibility via a separate 3.3V I/O power supply. For additional information about the XC4000XLA/XV device architecture, refer to the XC4000E/X FPGA Series general and functional descriptions.

- System-featured Field-Programmable Gate Arrays
  - Select-RAM™ memory: on-chip ultra-fast RAM with
    - Synchronous write option
    - Dual-port RAM option
  - Flexible function generators and abundant flip-flops
  - Dedicated high-speed carry logic
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution networks
- Flexible Array Architecture
- Low-power Segmented Routing Architecture
- Systems-oriented Features
  - IEEE 1149.1-compatible boundary scan
  - Individually programmable output slew rate
  - Programmable input pull-up or pull-down resistors
  - Unlimited reprogrammability
- Read Back Capability
  - Program verification and internal node observability

Electrical Features

- XLA Devices Require 3.0 - 3.6 V (VCC)
- XV Devices Require 2.3- 2.7 V (VCCINT) and 3.0 - 3.6 V (VCCIO)
- 5.0 V TTL compatible I/O
- 3.3 V LVTTL, LVCMOS compliant I/O
- 5.0 V and 3.0 V PCI Compliant I/O
- 12 mA or 24 mA Current Sink Capability
- Safe under All Power-up Sequences
- XLA Consumes 40% Less Power than XL
- XV Consumes 65% Less Power than XL
- Optional Input Clamping to VCC (XLA) or VCCIO (XV)

Additional Features

- Footprint Compatible with XC4000XL FPGAs - Lower cost with improved performance and lower power
- Advanced Technology — 5 layer metal, 0.25 µm CMOS process (XV) or 0.35 µm CMOS process (XLA)
- Highest Performance — System erformance beyond 100 MHz
- High Capacity — Up to 500,000 system gates and 270,000 synchronous SRAM bits
- Low Power — 3.3 V/2.5 V technology plus segmented routing architecture
- Safe and Easy to Use — Interfaces to any combination of 3.3 V and 5.0 V TTL compatible devices

Table 1: XC4000XLA Series Field Programmable Gate Arrays

Device	Logic Cells	Max Logic Gates (No RAM)	Max. RAM Bits (No Logic)	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. User I/O	Required Configuration Bits
XC4013XLA	1,368	13,000	18,432	10,000 - 30,000	24 x 24	576	1,536	192	393,632
XC4020XLA	1,862	20,000	25,088	13,000 - 40,000	28 x 28	784	2,016	224	521,880
XC4028XLA	2,432	28,000	32,768	18,000 - 50,000	32 x 32	1,024	2,560	256	668,184
XC4036XLA	3,078	36,000	41,472	22,000 - 65,000	36 x 36	1,296	3,168	288	832,528
XC4044XLA	3,800	44,000	51,200	27,000 - 80,000	40 x 40	1,600	3,840	320	1,014,928
XC4052XLA	4,598	52,000	61,952	33,000 - 100,000	44 x 44	1,936	4,576	352	1,215,368
XC4062XLA	5,472	62,000	73,728	40,000 - 130,000	48 x 48	2,304	5,376	384	1,433,864
XC4085XLA	7,448	85,000	100,352	55,000 - 180,000	56 x 56	3,136	7,168	448	1,924,992
XC40110XV	9,728	110,000	131,072	75,000 - 235,000	64 x 64	4,096	9,216	448	2,686,136
XC40150XV	12,312	150,000	165,888	100,000 - 300,000	72 x 72	5,184	11,520	448	3,373,448
XC40200XV	16,758	200,000	225,792	130,000 - 400,000	84 x 84	7,056	15,456	448	4,551,056
XC40250XV	20,102	250,000	270,848	180,000 - 500,000	92 x 92	8,464	18,400	448	5,433,888

* Maximum values of gate range assume 20-30% of CLBs used as RAM

Figure D.3: XC4020XLA FPGA Datasheet - Page 1 only

XILINX®

Spartan and SpartanXL Families Field Programmable Gate Arrays

January 6, 1999 (Version 1.4)

Preliminary Product Specification

Introduction

The Spartan™ Series is the first high-volume production FPGA solution to deliver all the key requirements for ASIC replacement up to 40,000 gates. These requirements include high performance, on-chip RAM, Core Solutions and prices that, in high volume, approach and in many cases are equivalent to mask programmed ASIC devices.

The Spartan Series is the result of more than thirteen years of FPGA design experience and feedback from thousands of customers. By streamlining the Spartan feature set, leveraging advanced hybrid process technologies and focusing on total cost management, the Spartan Series delivers the key features required by ASIC and other high volume logic users while avoiding the initial cost, long development cycles and inherent risk of conventional ASICs. The Spartan Series currently has 10 members, as shown in Table 1.

Spartan Series Features

Note: The Spartan Series devices described in this data sheet include the 5 V Spartan family of devices and the 3.3 V SpartanXL™ family of devices.

- Next generation ASIC replacement technology
  - First ASIC replacement FPGA for high-volume production with on-chip RAM
  - Advanced process technology
  - Density up to 1862 logic cells or 40,000 system gates
  - Streamlined feature set based on XC4000 architecture
  - System performance beyond 80 MHz
  - Broad set of AllianceCORE™ and LogiCORE™ pre-defined solutions available
  - Unlimited reprogrammability
  - Low cost
- System level features
  - Available in both 5.0 Volt and 3.3 Volt versions
  - On-chip SelectRAM™ memory
  - Fully PCI compliant
  - Low power segmented routing architecture
  - Full readback capability for program verification and internal node observability
  - Dedicated high-speed carry logic
  - Internal 3-state bus capability
  - 8 global low-skew clock or signal networks
  - IEEE 1149.1-compatible boundary scan logic
- Versatile I/O and packaging
  - Low cost plastic packages available in all densities
  - Footprint compatibility in common packages
  - Individually programmable output slew-rate control maximizes performance and reduces noise
  - Zero input register hold time simplifies system timing
- Fully supported by powerful Xilinx development system
  - Foundation series: Integrated, shrink-wrap software
  - Alliance series: Over 100 PC and workstation 3RD party development systems supported
  - Fully automatic mapping, placement and routing
  - Interactive design editor for design optimization

Additional SpartanXL Features

- 3.3V supply for low power with 5V tolerant I/Os
- Power down input
- Higher performance
- Faster carry logic
- More flexible high-speed clock network
- Latch capability in Configurable Logic Blocks
- Input fast capture latch
- Optional mux or 2-input function generator on outputs
- 12 mA or 24 mA output drive
- 5V/3.3V PCI compliant
- Enhanced Boundary Scan
- Express Mode configuration

Table 1: Spartan and SpartanXL Series Field Programmable Gate Arrays

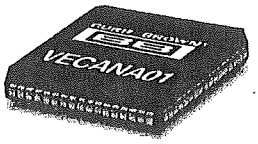
Device	Logic Cells	Max System Gates	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. Available User I/O
XCS05 & XCS05XL	238	5,000	2,000 - 5,000	10 x 10	100	360	77
XCS10 & XCS10XL	466	10,000	3,000 - 10,000	14 x 14	196	616	112
XCS20 & XCS20XL	950	20,000	7,000 - 20,000	20 x 20	400	1,120	160
XCS30 & XCS30XL	1368	30,000	10,000 - 30,000	24 x 24	576	1,536	192
XCS40 & XCS40XL	1862	40,000	13,000 - 40,000	28 x 28	784	2,016	205

* Max values of Typical Gate Range include 20-30% of CLBs used as RAM.

Figure D.4: Spartan FPGA Datasheet - Page 1 only

Burr-Brown Products
from Texas Instruments

VECANA01



www.ti.com

10-Channel, 12-Bit DATA ACQUISITION SYSTEM

FEATURES

- 10 FULLY DIFFERENTIAL INPUTS
- 5 SIMULTANEOUS SAMPLED CHANNELS PLUS 2 SYNCHRONIZED SAMPLING CHANNELS
- 3 SYNCHRONIZED 12-BIT ADCs
- 12.8µs THROUGHPUT RATE
- DIGITALLY SELECTABLE INPUT RANGES
- ±5V POWER SUPPLIES
- SERIAL DIGITAL INPUT/OUTPUTS
- 7 SIGN AND 3 DIGITALLY PROGRAMMABLE WINDOW COMPARATOR

APPLICATIONS

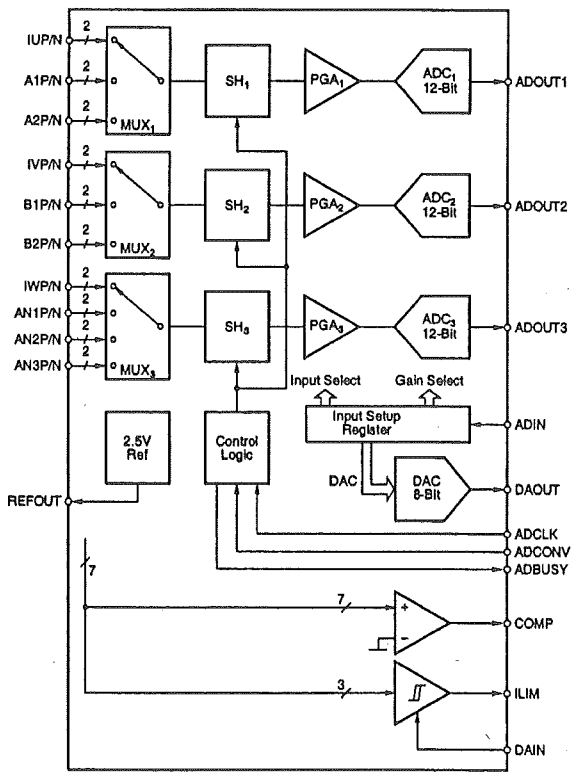
- AC MOTOR SPEED CONTROLS
- THREE PHASE POWER CONTROL
- UNINTERRUPTABLE POWER SUPPLIES
- VIBRATION ANALYSIS

DESCRIPTION

The VECANA01 consists of three 12-bit analog-to-digital converters preceded by five simultaneously operating sample-hold amplifiers, and multiplexers for 10 differential inputs. The ADCs have simultaneous serial outputs for high speed data transfer and data processing.

The VECANA01 also offers a programmable gain amplifier with programmable gains of 1.0V/V, 1.25V/V, 2.5V/V, and 5.0V/V. Channel selection and gain selection are selectable through the serial input control word. The high through put rate is maintained by simultaneously clocking in the 13-bit input control word for the next conversion while the present conversions are clocked out.

The part also contains an 8-bit digital-to-analog converter whose digital input is supplied as part of the input control word.



TEXAS INSTRUMENTS

Figure D.5: VECANA01 Datasheet - Page 1 only

Appendix E. Motor Parameters

The induction motor used for testing was a high-speed three-phase induction motor with a integrated reduction gear. The parameters taken from the nameplate are shown in Table E.1. The DC resistance of the stator windings was measured using a Wheatstone bridge. The phase-to-phase resistance was 0.035Ω, which corresponds to a per-phase stator resistance of 0.0175Ω.

The parameters of the motor were obtained by performing no-load and blocked rotor tests. For the no-load tests, the reduction gear was removed in an attempt to reduce the load on the rotor. There was still a relatively significant load due to the cooling fins on the rotor that move a large amount of air due to the high rotor speed. Later testing with the PDL motor drive and load motor showed 1.8kW was required to spin the test motor at full-speed. Better no-load test results could be obtained if the load motor was used to spin the rotor at synchronous speed while the no-load measurements were performed.

The motor was driven by the three-level inverter, which was configured to produce a sinusoidal voltage with a fixed amplitude and frequency. The applied voltage and frequency, and the corresponding current and power factor are listed in Table E.2 and Table E.3 for the no-load and blocked rotor tests respectively. The induction motor parameters were calculated from these tests and the results are shown in Table E.4.

Table E.1: Test Motor Nameplate Data

Parameter	Value
Power	15hp
Voltage	200v
Current	55.6A
Frequency	400Hz
Number of Poles (P)	4
Motor Speed	11,500RPM
Gear Ratio	4.6667:1
Shaft Speed	2,470RPM
Temp Rise (Continuous Duty)	50deg C

Table E.2: No-Load Test Results

Parameter	Value
Applied Frequency	300Hz
Applied Voltage (phase)	84.9V rms
Current (phase)	25A rms
Power Factor	0.174

Table E.3: Blocked Rotor Test Results

Parameter	Value
Applied Frequency	400Hz
Applied Voltage (phase)	48.99V rms
Current (phase)	40A rms
Power Factor	0.669

Table E.4: Motor Parameters

Parameter	Value
Rated Power	15hp
Rated Voltage	200v
Rated Current	55.6A
Rated Frequency	400Hz
Number of Poles (P)	4
Stator Resistance (r_s)	0.0175Ω
Stator Self-Inductance (L_s)	2.01mH
Magnetising Inductance (L_m)	1.83mH
Rotor Resistance (r_r)	0.802Ω
Rotor Self-Inductance (L_r)	2.01mH